
RsCmwCdma2kMeas

Release 3.8.10.8

Rohde & Schwarz

May 27, 2021

CONTENTS:

1	Getting Started	3
1.1	Introduction	3
1.2	Installation	5
1.3	Finding Available Instruments	6
1.4	Initiating Instrument Session	6
1.5	Plain SCPI Communication	10
1.6	Error Checking	12
1.7	Exception Handling	12
1.8	Transferring Files	14
1.9	Writing Binary Data	14
1.10	Transferring Big Data with Progress	15
1.11	Multithreading	16
2	Revision History	21
3	Enums	23
3.1	BandClass	23
3.2	CmwsConnector	23
3.3	ParameterSetMode	23
3.4	Rbw	24
3.5	Rconfig	24
3.6	Repeat	24
3.7	ResourceState	24
3.8	ResultStatus2	24
3.9	RetriggerMode	25
3.10	RetriggerOption	25
3.11	RxConnector	25
3.12	RxConverter	26
3.13	Scenario	26
3.14	SigChStateA	26
3.15	SigChStateB	26
3.16	Slope	26
3.17	SpreadingFactor	27
3.18	StatePower	27
3.19	StopCondition	27
3.20	Tab	27
3.21	UpDownDirection	27
4	RepCaps	29
4.1	Instance (Global)	29

4.2	AcpMinus	29
4.3	AcpPlus	29
4.4	Segment	30
4.5	Sequence	30
5	Examples	31
6	Index	33
7	RsCmwCdma2kMeas API Structure	35
7.1	Route	37
7.1.1	Scenario	37
7.1.1.1	Catalog	39
7.2	Configure	39
7.2.1	RfSettings	41
7.2.2	MultiEval	45
7.2.2.1	Scount	48
7.2.2.2	ListPy	49
7.2.2.2.1	SingleCmw	50
7.2.2.2.2	Segment<Segment>	51
7.2.2.2.2.1	Setup	51
7.2.2.2.2.2	SingleCmw	53
7.2.2.2.2.3	Connector	53
7.2.2.2.2.4	Modulation	54
7.2.2.2.2.5	Spectrum	55
7.2.2.3	Acp	56
7.2.2.3.1	Extended	57
7.2.2.4	Result	58
7.2.2.5	Limit	66
7.2.2.5.1	Acp	73
7.2.3	Oltr	74
7.2.3.1	Pstep	76
7.2.3.2	RpInterval	77
7.2.3.3	Ginterval	78
7.2.3.4	Limit	78
7.3	Trigger	79
7.3.1	MultiEval	79
7.3.1.1	Catalog	82
7.3.1.2	ListPy	83
7.4	MultiEval	84
7.4.1	State	86
7.4.1.1	All	87
7.4.2	Trace	88
7.4.2.1	EvMagnitude	88
7.4.2.1.1	Current	88
7.4.2.1.2	Average	89
7.4.2.1.3	Maximum	89
7.4.2.2	Merror	90
7.4.2.2.1	Current	90
7.4.2.2.2	Average	91
7.4.2.2.3	Maximum	92
7.4.2.3	Perror	92
7.4.2.3.1	Current	93
7.4.2.3.2	Average	93

7.4.2.3.3	Maximum	94
7.4.2.4	Acp	95
7.4.2.4.1	Current	95
7.4.2.4.1.1	Relative	95
7.4.2.4.1.2	Absolute	96
7.4.2.4.2	Extended	98
7.4.2.4.2.1	Current	98
7.4.2.4.2.2	Relative	98
7.4.2.4.2.3	Absolute	99
7.4.2.4.2.4	Average	101
7.4.2.4.2.5	Relative	101
7.4.2.4.2.6	Absolute	102
7.4.2.4.2.7	Maximum	104
7.4.2.4.2.8	Relative	104
7.4.2.4.2.9	Absolute	105
7.4.2.4.3	Average	107
7.4.2.4.3.1	Relative	107
7.4.2.4.3.2	Absolute	108
7.4.2.4.4	Maximum	109
7.4.2.4.4.1	Relative	110
7.4.2.4.4.2	Absolute	111
7.4.2.5	Obw	112
7.4.2.5.1	Current	112
7.4.2.6	Spectrum	113
7.4.2.6.1	Current	114
7.4.2.7	Cdp	115
7.4.2.7.1	Isignal	115
7.4.2.7.1.1	Current	115
7.4.2.7.1.2	Average	116
7.4.2.7.1.3	Maximum	117
7.4.2.7.1.4	Minimum	119
7.4.2.7.1.5	State	120
7.4.2.7.1.6	Limit	120
7.4.2.7.2	Qsignal	121
7.4.2.7.2.1	Current	121
7.4.2.7.2.2	Average	122
7.4.2.7.2.3	Maximum	123
7.4.2.7.2.4	Minimum	124
7.4.2.7.2.5	State	125
7.4.2.7.2.6	Limit	126
7.4.2.8	Cde	126
7.4.2.8.1	Isignal	126
7.4.2.8.1.1	Current	127
7.4.2.8.1.2	Average	128
7.4.2.8.1.3	Maximum	129
7.4.2.8.1.4	State	130
7.4.2.8.1.5	Limit	131
7.4.2.8.2	Qsignal	131
7.4.2.8.2.1	Current	131
7.4.2.8.2.2	Average	132
7.4.2.8.2.3	Maximum	134
7.4.2.8.2.4	State	135
7.4.2.8.2.5	Limit	135
7.4.2.9	Cp	136

7.4.2.9.1	Isignal	136
7.4.2.9.1.1	Current	136
7.4.2.9.1.2	Average	137
7.4.2.9.1.3	Maximum	139
7.4.2.9.1.4	Minimum	140
7.4.2.9.1.5	State	141
7.4.2.9.2	Qsignal	142
7.4.2.9.2.1	Current	142
7.4.2.9.2.2	Average	143
7.4.2.9.2.3	Maximum	145
7.4.2.9.2.4	Minimum	147
7.4.2.9.2.5	State	148
7.4.2.10	Cpo	149
7.4.2.10.1	Isignal	149
7.4.2.10.1.1	Current	149
7.4.2.10.1.2	Average	151
7.4.2.10.1.3	Maximum	152
7.4.2.10.1.4	State	153
7.4.2.10.2	Qsignal	154
7.4.2.10.2.1	Current	154
7.4.2.10.2.2	Average	156
7.4.2.10.2.3	Maximum	157
7.4.2.10.2.4	State	159
7.4.2.11	Cto	159
7.4.2.11.1	Isignal	160
7.4.2.11.1.1	Current	160
7.4.2.11.1.2	Average	161
7.4.2.11.1.3	Maximum	163
7.4.2.11.1.4	State	164
7.4.2.11.2	Qsignal	164
7.4.2.11.2.1	Current	165
7.4.2.11.2.2	Average	166
7.4.2.11.2.3	Maximum	168
7.4.2.11.2.4	State	169
7.4.2.12	Iq	170
7.4.2.12.1	Current	170
7.4.3	Modulation	171
7.4.3.1	Current	171
7.4.3.2	Average	174
7.4.3.3	Maximum	177
7.4.3.4	Minimum	179
7.4.3.5	StandardDev	181
7.4.4	Acp	184
7.4.4.1	Current	185
7.4.4.2	Average	186
7.4.4.3	Maximum	188
7.4.5	Obw	189
7.4.5.1	Current	189
7.4.5.2	Average	190
7.4.5.3	Maximum	192
7.4.6	ListPy	193
7.4.6.1	Sreliability	193
7.4.6.2	Modulation	194
7.4.6.2.1	Otolerance	194

7.4.6.2.2	StCount	195
7.4.6.2.3	Evm	195
7.4.6.2.3.1	Rms	195
7.4.6.2.3.2	Current	196
7.4.6.2.3.3	Average	196
7.4.6.2.3.4	Maximum	197
7.4.6.2.3.5	StandardDev	198
7.4.6.2.3.6	Peak	199
7.4.6.2.3.7	Current	199
7.4.6.2.3.8	Average	200
7.4.6.2.3.9	Maximum	201
7.4.6.2.3.10	StandardDev	201
7.4.6.2.4	Merror	202
7.4.6.2.4.1	Rms	202
7.4.6.2.4.2	Current	203
7.4.6.2.4.3	Average	204
7.4.6.2.4.4	Maximum	204
7.4.6.2.4.5	StandardDev	205
7.4.6.2.4.6	Peak	206
7.4.6.2.4.7	Current	206
7.4.6.2.4.8	Average	207
7.4.6.2.4.9	Maximum	208
7.4.6.2.4.10	StandardDev	209
7.4.6.2.5	Perror	209
7.4.6.2.5.1	Rms	210
7.4.6.2.5.2	Current	210
7.4.6.2.5.3	Average	211
7.4.6.2.5.4	Maximum	212
7.4.6.2.5.5	StandardDev	212
7.4.6.2.5.6	Peak	213
7.4.6.2.5.7	Current	213
7.4.6.2.5.8	Average	214
7.4.6.2.5.9	Maximum	215
7.4.6.2.5.10	StandardDev	216
7.4.6.2.6	IqOffset	217
7.4.6.2.6.1	Current	217
7.4.6.2.6.2	Average	218
7.4.6.2.6.3	Maximum	218
7.4.6.2.6.4	StandardDev	219
7.4.6.2.7	IqImbalance	220
7.4.6.2.7.1	Current	220
7.4.6.2.7.2	Average	221
7.4.6.2.7.3	Maximum	222
7.4.6.2.7.4	StandardDev	223
7.4.6.2.8	FreqError	223
7.4.6.2.8.1	Current	224
7.4.6.2.8.2	Average	225
7.4.6.2.8.3	Maximum	225
7.4.6.2.8.4	StandardDev	226
7.4.6.2.9	Terror	227
7.4.6.2.9.1	Current	227
7.4.6.2.9.2	Average	228
7.4.6.2.9.3	Maximum	229
7.4.6.2.9.4	StandardDev	230

7.4.6.2.10	Wquality	230
7.4.6.2.10.1	Current	231
7.4.6.2.10.2	Pmax	232
7.4.6.2.10.3	Current	232
7.4.6.2.10.4	Pmin	233
7.4.6.2.10.5	Current	233
7.4.6.2.10.6	Average	234
7.4.6.2.10.7	Maximum	235
7.4.6.2.10.8	StandardDev	235
7.4.6.2.11	PwBand	236
7.4.6.2.11.1	Current	236
7.4.6.2.11.2	Average	237
7.4.6.2.11.3	Maximum	238
7.4.6.2.11.4	Minimum	239
7.4.6.2.11.5	StandardDev	240
7.4.6.2.12	PnBand	241
7.4.6.2.12.1	Current	241
7.4.6.2.12.2	Average	242
7.4.6.2.12.3	Maximum	243
7.4.6.2.12.4	Minimum	243
7.4.6.2.12.5	StandardDev	244
7.4.6.2.13	Current	245
7.4.6.2.14	Average	248
7.4.6.2.15	Maximum	250
7.4.6.2.16	Minimum	253
7.4.6.2.17	StandardDev	255
7.4.6.3	Segment<Segment>	258
7.4.6.3.1	Modulation	258
7.4.6.3.1.1	Current	258
7.4.6.3.1.2	Average	261
7.4.6.3.1.3	Maximum	263
7.4.6.3.1.4	Minimum	266
7.4.6.3.1.5	StandardDev	268
7.4.6.3.2	Acp	271
7.4.6.3.2.1	Current	271
7.4.6.3.2.2	Extended	273
7.4.6.3.2.3	Current	273
7.4.6.3.2.4	Average	275
7.4.6.3.2.5	Maximum	276
7.4.6.3.2.6	Minimum	278
7.4.6.3.2.7	StandardDev	280
7.4.6.3.2.8	Average	282
7.4.6.3.2.9	Maximum	283
7.4.6.3.2.10	Minimum	285
7.4.6.3.2.11	StandardDev	287
7.4.6.3.3	Obw	289
7.4.6.3.3.1	Current	289
7.4.6.3.3.2	Average	290
7.4.6.3.3.3	Maximum	292
7.4.6.3.3.4	StandardDev	293
7.4.6.3.4	Cp	295
7.4.6.3.4.1	Current	295
7.4.6.3.4.2	Average	297
7.4.6.3.4.3	Maximum	299

7.4.6.3.4.4	Minimum	301
7.4.6.3.4.5	State	303
7.4.6.3.5	Cpo	304
7.4.6.3.5.1	Current	304
7.4.6.3.5.2	Average	306
7.4.6.3.5.3	Maximum	308
7.4.6.3.5.4	State	310
7.4.6.3.6	Cto	311
7.4.6.3.6.1	Current	311
7.4.6.3.6.2	Average	313
7.4.6.3.6.3	Maximum	315
7.4.6.3.6.4	State	317
7.4.6.4	Acp	318
7.4.6.4.1	Otolerance	319
7.4.6.4.2	StCount	319
7.4.6.4.3	Acpm<AcpMinus>	320
7.4.6.4.3.1	Current	320
7.4.6.4.3.2	Average	321
7.4.6.4.3.3	Maximum	322
7.4.6.4.4	Extended	323
7.4.6.4.4.1	Acpm<AcpMinus>	323
7.4.6.4.4.2	Current	323
7.4.6.4.4.3	Average	324
7.4.6.4.4.4	Maximum	325
7.4.6.4.4.5	Acpp<AcpPlus>	326
7.4.6.4.4.6	Current	327
7.4.6.4.4.7	Average	328
7.4.6.4.4.8	Maximum	329
7.4.6.4.4.9	Current	330
7.4.6.4.4.10	Average	331
7.4.6.4.4.11	Maximum	333
7.4.6.4.4.12	Minimum	335
7.4.6.4.4.13	StandardDev	336
7.4.6.4.5	Acpp<AcpPlus>	338
7.4.6.4.5.1	Current	338
7.4.6.4.5.2	Average	339
7.4.6.4.5.3	Maximum	340
7.4.6.4.6	Npow	341
7.4.6.4.6.1	Current	341
7.4.6.4.6.2	Average	342
7.4.6.4.6.3	Maximum	343
7.4.6.4.6.4	Minimum	344
7.4.6.4.6.5	StandardDev	344
7.4.6.4.7	Wpow	345
7.4.6.4.7.1	Current	345
7.4.6.4.7.2	Average	346
7.4.6.4.7.3	Maximum	347
7.4.6.4.7.4	Minimum	348
7.4.6.4.7.5	StandardDev	348
7.4.6.4.8	Current	349
7.4.6.4.9	Average	352
7.4.6.4.10	Maximum	355
7.4.6.4.11	Minimum	358
7.4.6.4.12	StandardDev	360

7.4.6.5	Obw	363
7.4.6.5.1	Frequency	363
7.4.6.5.1.1	Current	364
7.4.6.5.1.2	Average	365
7.4.6.5.1.3	Maximum	365
7.4.6.5.1.4	StandardDev	366
7.4.6.5.1.5	Lower	367
7.4.6.5.1.6	Upper	367
7.4.6.5.2	Current	368
7.4.6.5.3	Average	369
7.4.6.5.4	Maximum	371
7.4.6.5.5	StandardDev	372
7.4.6.6	Cp	373
7.4.6.6.1	RpiCh	374
7.4.6.6.1.1	State	374
7.4.6.6.1.2	Current	374
7.4.6.6.1.3	Average	375
7.4.6.6.1.4	Maximum	376
7.4.6.6.1.5	Minimum	377
7.4.6.6.2	RdcCh	377
7.4.6.6.2.1	State	378
7.4.6.6.2.2	Current	378
7.4.6.6.2.3	Average	379
7.4.6.6.2.4	Maximum	380
7.4.6.6.2.5	Minimum	380
7.4.6.6.3	RccCh	381
7.4.6.6.3.1	State	381
7.4.6.6.3.2	Current	382
7.4.6.6.3.3	Average	382
7.4.6.6.3.4	Maximum	383
7.4.6.6.3.5	Minimum	384
7.4.6.6.4	ReaCh	385
7.4.6.6.4.1	State	385
7.4.6.6.4.2	Current	385
7.4.6.6.4.3	Average	386
7.4.6.6.4.4	Maximum	387
7.4.6.6.4.5	Minimum	387
7.4.6.6.5	Rfch	388
7.4.6.6.5.1	State	388
7.4.6.6.5.2	Current	389
7.4.6.6.5.3	Average	390
7.4.6.6.5.4	Maximum	390
7.4.6.6.5.5	Minimum	391
7.4.6.6.6	RsCh	392
7.4.6.6.6.1	Ztef	392
7.4.6.6.6.2	State	392
7.4.6.6.6.3	Current	393
7.4.6.6.6.4	Average	394
7.4.6.6.6.5	Maximum	394
7.4.6.6.6.6	Minimum	395
7.4.6.6.6.7	Zoet	396
7.4.6.6.6.8	State	396
7.4.6.6.6.9	Current	397
7.4.6.6.6.10	Average	397

7.4.6.6.6.11	Maximum	398
7.4.6.6.6.12	Minimum	399
7.4.6.6.7	Current	400
7.4.6.6.8	Average	402
7.4.6.6.9	Maximum	404
7.4.6.6.10	Minimum	406
7.4.6.6.11	State	408
7.4.6.7	Cpo	409
7.4.6.7.1	RpiCh	409
7.4.6.7.1.1	State	409
7.4.6.7.1.2	Current	410
7.4.6.7.1.3	Average	410
7.4.6.7.1.4	Maximum	411
7.4.6.7.2	RdcCh	412
7.4.6.7.2.1	State	412
7.4.6.7.2.2	Current	412
7.4.6.7.2.3	Average	413
7.4.6.7.2.4	Maximum	414
7.4.6.7.3	RccCh	415
7.4.6.7.3.1	State	415
7.4.6.7.3.2	Current	415
7.4.6.7.3.3	Average	416
7.4.6.7.3.4	Maximum	417
7.4.6.7.4	ReaCh	417
7.4.6.7.4.1	State	418
7.4.6.7.4.2	Current	418
7.4.6.7.4.3	Average	419
7.4.6.7.4.4	Maximum	420
7.4.6.7.5	Rfch	420
7.4.6.7.5.1	State	421
7.4.6.7.5.2	Current	421
7.4.6.7.5.3	Average	422
7.4.6.7.5.4	Maximum	423
7.4.6.7.6	RsCh	423
7.4.6.7.6.1	Ztef	424
7.4.6.7.6.2	State	424
7.4.6.7.6.3	Current	424
7.4.6.7.6.4	Average	425
7.4.6.7.6.5	Maximum	426
7.4.6.7.6.6	Zoet	427
7.4.6.7.6.7	State	427
7.4.6.7.6.8	Current	427
7.4.6.7.6.9	Average	428
7.4.6.7.6.10	Maximum	429
7.4.6.7.7	Current	430
7.4.6.7.8	Average	432
7.4.6.7.9	Maximum	434
7.4.6.7.10	State	436
7.4.6.8	Cto	437
7.4.6.8.1	RpiCh	437
7.4.6.8.1.1	State	437
7.4.6.8.1.2	Current	438
7.4.6.8.1.3	Average	438
7.4.6.8.1.4	Maximum	439

7.4.6.8.2	RdcCh	440
7.4.6.8.2.1	State	440
7.4.6.8.2.2	Current	440
7.4.6.8.2.3	Average	441
7.4.6.8.2.4	Maximum	442
7.4.6.8.3	RccCh	442
7.4.6.8.3.1	State	443
7.4.6.8.3.2	Current	443
7.4.6.8.3.3	Average	444
7.4.6.8.3.4	Maximum	445
7.4.6.8.4	ReaCh	445
7.4.6.8.4.1	State	446
7.4.6.8.4.2	Current	446
7.4.6.8.4.3	Average	447
7.4.6.8.4.4	Maximum	448
7.4.6.8.5	Rfch	448
7.4.6.8.5.1	State	449
7.4.6.8.5.2	Current	449
7.4.6.8.5.3	Average	450
7.4.6.8.5.4	Maximum	451
7.4.6.8.6	RsCh	451
7.4.6.8.6.1	Ztef	452
7.4.6.8.6.2	State	452
7.4.6.8.6.3	Current	452
7.4.6.8.6.4	Average	453
7.4.6.8.6.5	Maximum	454
7.4.6.8.6.6	Zoet	455
7.4.6.8.6.7	State	455
7.4.6.8.6.8	Current	455
7.4.6.8.6.9	Average	456
7.4.6.8.6.10	Maximum	457
7.4.6.8.7	Current	458
7.4.6.8.8	Average	460
7.4.6.8.9	Maximum	462
7.4.6.8.10	State	464
7.5	Oltr	465
7.5.1	State	467
7.5.1.1	All	468
7.5.2	Sequence<Sequence>	469
7.5.2.1	Trace	469
7.5.2.1.1	Up	469
7.5.2.1.1.1	State	470
7.5.2.1.2	Down	472
7.5.2.1.2.1	State	473
7.6	RpInterval	474
7.6.1	Sequence<Sequence>	474
7.6.1.1	Trace	475
7.6.1.1.1	Up	475
7.6.1.1.2	Down	476



GETTING STARTED

1.1 Introduction



RsCmwCdma2kMeas is a Python remote-control communication module for Rohde & Schwarz SCPI-based Test and Measurement Instruments. It represents SCPI commands as fixed APIs and hence provides SCPI autocompletion and helps you to avoid common string typing mistakes.

Basic example of the idea:

SCPI command:

SYSTem:REFeRence:FREQuency:SOURce

Python module representation:

writing:

```
driver.system.reference.frequency.source.set()
```

reading:

```
driver.system.reference.frequency.source.get()
```

Check out this RsCmwBase example:

```
""" Example on how to use the python RsCmw auto-generated instrument driver showing:
- usage of basic properties of the cmw_base object
- basic concept of setting commands and repcaps: DISPlay:WINDow<n>:SElect
- cmw_xxx drivers reliability interface usage
"""

from RsCmwBase import * # install from pypi.org

RsCmwBase.assert_minimum_version('3.7.90.32')
cmw_base = RsCmwBase('TCPIP::10.112.1.116::INSTR', True, False)
print(f'CMW Base IND: {cmw_base.utilities.idn_string}')
print(f'CMW Instrument options:\n{" ".join(cmw_base.utilities.instrument_options)}')
cmw_base.utilities.visa_timeout = 5000

# Sends OPC after each command
cmw_base.utilities.opc_query_after_write = False
```

(continues on next page)

(continued from previous page)

```

# Checks for syst:err? after each command / query
cmw_base.utilities.instrument_status_checking = True

# DISPLAY:WINDOW<n>:SELECT
cmw_base.display.window.select.set(repcap.Window.Win1)
cmw_base.display.window.repcap_window_set(repcap.Window.Win2)
cmw_base.display.window.select.set()

# Self-test
self_test = cmw_base.utilities.self_test()
print(f'CMW self-test result: {self_test} - {"Passed" if self_test[0] == 0 else "Failed"}'
      ↪ '')

# Driver's Interface reliability offers a convenient way of reacting on the return value ↪
↪ Reliability Indicator
cmw_base.reliability.ExceptionOnError = True

# Callback to use for the reliability indicator update event
def my_reliability_handler(event_args: ReliabilityEventArgs):
    print(f'Base Reliability updated.\nContext: {event_args.context}\nMessage:
    ↪ {event_args.message}')

# We register a callback for each change in the reliability indicator
cmw_base.reliability.on_update_handler = my_reliability_handler

# You can obtain the last value of the returned reliability
print(f"\nReliability last value: {cmw_base.reliability.last_value}, context '{cmw_base.
    ↪ reliability.last_context}', message: {cmw_base.reliability.last_message}")

# Reference Frequency Source
cmw_base.system.reference.frequency.source_set(enums.SourceIntExt.INTERNAL)

# Close the session
cmw_base.close()

```

Couple of reasons why to choose this module over plain SCPI approach:

- Type-safe API using typing module
- You can still use the plain SCPI communication
- You can select which VISA to use or even not use any VISA at all
- Initialization of a new session is straight-forward, no need to set any other properties
- Many useful features are already implemented - reset, self-test, opc-synchronization, error checking, option checking
- Binary data blocks transfer in both directions
- Transfer of arrays of numbers in binary or ASCII format
- File transfers in both directions
- Events generation in case of error, sent data, received data, chunk data (in case of big data transfer)

- Multithreading session locking - you can use multiple threads talking to one instrument at the same time

1.2 Installation

RsCmwCdma2kMeas is hosted on pypi.org. You can install it with pip (for example, `pip.exe` for Windows), or if you are using Pycharm (and you should be :) direct in the Pycharm **Package Management** GUI.

Preconditions

- Installed VISA. You can skip this if you plan to use only socket LAN connection. Download the Rohde & Schwarz VISA for Windows, Linux, Mac OS from [here](#)

Option 1 - Installing with pip.exe under Windows

- Start the command console: WinKey + R, type `cmd` and hit ENTER
- Change the working directory to the Python installation of your choice (adjust the user name and python version in the path):

```
cd c:\Users\John\AppData\Local\Programs\Python\Python37\Scripts
```

- Install with the command: `pip install RsCmwCdma2kMeas`

Option 2 - Installing in Pycharm

- In Pycharm Menu **File->Settings->Project->Project Interpreter** click on the '+' button on the bottom left
- Type `RsCmwCdma2kMeas` in the search box
- If you are behind a Proxy server, configure it in the Menu: **File->Settings->Appearance->System Settings->HTTP Proxy**

For more information about Rohde & Schwarz instrument remote control, check out our [Instrument Remote Control Web Series](#).

Option 3 - Offline Installation

If you are still reading the installation chapter, it is probably because the options above did not work for you - proxy problems, your boss saw the internet bill... Here are 5 easy steps for installing the RsCmwCdma2kMeas offline:

- Download this python script (**Save target as**): `rsinstrument_offline_install.py` This installs all the preconditions that the RsCmwCdma2kMeas needs.
- Execute the script in your offline computer (supported is python 3.6 or newer)
- Download the RsCmwCdma2kMeas package to your computer from the pypi.org: <https://pypi.org/project/RsCmwCdma2kMeas/#files> for example `c:\temp\`
- Start the command line WinKey + R, type `cmd` and hit ENTER
- Change the working directory to the Python installation of your choice (adjust the user name and python version in the path):

```
cd c:\Users\John\AppData\Local\Programs\Python\Python37\Scripts
```

- Install with the command: `pip install c:\temp\RsCmwCdma2kMeas-3.8.10.8.tar`

1.3 Finding Available Instruments

Like the pyvisa's ResourceManager, the RsCmwCdma2kMeas can search for available instruments:

```
"""
Find the instruments in your environment
"""

from RsCmwCdma2kMeas import *

# Use the instr_list string items as resource names in the RsCmwCdma2kMeas constructor
instr_list = RsCmwCdma2kMeas.list_resources("?*")
print(instr_list)
```

If you have more VISAs installed, the one actually used by default is defined by a secret widget called Visa Conflict Manager. You can force your program to use a VISA of your choice:

```
"""
Find the instruments in your environment with the defined VISA implementation
"""

from RsCmwCdma2kMeas import *

# In the optional parameter visa_select you can use for example 'rs' or 'ni'
# Rs Visa also finds any NRP-Zxx USB sensors
instr_list = RsCmwCdma2kMeas.list_resources('?*', 'rs')
print(instr_list)
```

Tip: We believe our R&S VISA is the best choice for our customers. Here are the reasons why:

- Small footprint
 - Superior VXI-11 and HiSLIP performance
 - Integrated legacy sensors NRP-Zxx support
 - Additional VXI-11 and LXI devices search
 - Availability for Windows, Linux, Mac OS
-

1.4 Initiating Instrument Session

RsCmwCdma2kMeas offers four different types of starting your remote-control session. We begin with the most typical case, and progress with more special ones.

Standard Session Initialization

Initiating new instrument session happens, when you instantiate the RsCmwCdma2kMeas object. Below, is a simple Hello World example. Different resource names are examples for different physical interfaces.

```
"""
Simple example on how to use the RsCmwCdma2kMeas module for remote-controlling your_
↳instrument
Preconditions:

- Installed RsCmwCdma2kMeas Python module Version 3.8.10 or newer from pypi.org
- Installed VISA, for example R&S Visa 5.12 or newer
"""

from RsCmwCdma2kMeas import *

# A good practice is to assure that you have a certain minimum version installed
RsCmwCdma2kMeas.assert_minimum_version('3.8.10')
resource_string_1 = 'TCPIP::192.168.2.101::INSTR' # Standard LAN connection (also_
↳called VXI-11)
resource_string_2 = 'TCPIP::192.168.2.101::hislip0' # Hi-Speed LAN connection - see_
↳1MA208
resource_string_3 = 'GPIB::20::INSTR' # GPIB Connection
resource_string_4 = 'USB::0x0AAD::0x0119::022019943::INSTR' # USB-TMC (Test and_
↳Measurement Class)

# Initializing the session
driver = RsCmwCdma2kMeas(resource_string_1)

idn = driver.utilities.query_str('*IDN?')
print(f"\nHello, I am: '{idn}'")
print(f'RsCmwCdma2kMeas package version: {driver.utilities.driver_version}')
print(f'Visa manufacturer: {driver.utilities.visa_manufacturer}')
print(f'Instrument full name: {driver.utilities.full_instrument_model_name}')
print(f'Instrument installed options: {",".join(driver.utilities.instrument_options)}')

# Close the session
driver.close()
```

Note: If you are wondering about the missing ASRL1::INSTR, yes, it works too, but come on... it's 2021.

Do not care about specialty of each session kind; RsCmwCdma2kMeas handles all the necessary session settings for you. You immediately have access to many identification properties in the interface `driver.utilities`. Here are some of them:

- `idn_string`
- `driver_version`
- `visa_manufacturer`
- `full_instrument_model_name`
- `instrument_serial_number`
- `instrument_firmware_version`

- instrument_options

The constructor also contains optional boolean arguments `id_query` and `reset`:

```
driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::HISLIP', id_query=True, reset=True)
```

- Setting `id_query` to `True` (default is `True`) checks, whether your instrument can be used with the `RsCmwCdma2kMeas` module.
- Setting `reset` to `True` (default is `False`) resets your instrument. It is equivalent to calling the `reset()` method.

Selecting a Specific VISA

Just like in the function `list_resources()`, the `RsCmwCdma2kMeas` allows you to choose which VISA to use:

```
"""
Choosing VISA implementation
"""

from RsCmwCdma2kMeas import *

# Force use of the Rs Visa. For NI Visa, use the "SelectVisa='ni'"
driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR', True, True, "SelectVisa='rs'")

idn = driver.utilities.query_str('*IDN?')
print(f"\nHello, I am: '{idn}'")
print(f"\nI am using the VISA from: {driver.utilities.visa_manufacturer}")

# Close the session
driver.close()
```

No VISA Session

We recommend using VISA when possible preferably with HiSlip session because of its low latency. However, if you are a strict VISA denier, `RsCmwCdma2kMeas` has something for you too - **no Visa installation raw LAN socket**:

```
"""
Using RsCmwCdma2kMeas without VISA for LAN Raw socket communication
"""

from RsCmwCdma2kMeas import *

driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::5025::SOCKET', True, True, "SelectVisa=
↪ 'socket'")
print(f'Visa manufacturer: {driver.utilities.visa_manufacturer}')
print(f"\nHello, I am: '{driver.utilities.idn_string}'")

# Close the session
driver.close()
```

Warning: Not using VISA can cause problems by debugging when you want to use the communication Trace Tool. The good news is, you can easily switch to use VISA and back just by changing the constructor arguments. The rest of your code stays unchanged.

Simulating Session

If a colleague is currently occupying your instrument, leave him in peace, and open a simulating session:

```
driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::HISLIP', True, True, "Simulate=True")
```

More option_string tokens are separated by comma:

```
driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::HISLIP', True, True, "SelectVisa='rs',  
↪Simulate=True")
```

Shared Session

In some scenarios, you want to have two independent objects talking to the same instrument. Rather than opening a second VISA connection, share the same one between two or more RsCmwCdma2kMeas objects:

```
"""
Sharing the same physical VISA session by two different RsCmwCdma2kMeas objects
"""

from RsCmwCdma2kMeas import *

driver1 = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR', True, True)
driver2 = RsCmwCdma2kMeas.from_existing_session(driver1)

print(f'driver1: {driver1.utilities.idn_string}')
print(f'driver2: {driver2.utilities.idn_string}')

# Closing the driver2 session does not close the driver1 session - driver1 is the
↪ 'session master'
driver2.close()
print(f'driver2: I am closed now')

print(f'driver1: I am still opened and working: {driver1.utilities.idn_string}')
driver1.close()
print(f'driver1: Only now I am closed.')
```

Note: The driver1 is the object holding the ‘master’ session. If you call the driver1.close(), the driver2 loses its instrument session as well, and becomes pretty much useless.

1.5 Plain SCPI Communication

After you have opened the session, you can use the instrument-specific part described in the RsCmwCdma2kMeas API Structure. If for any reason you want to use the plain SCPI, use the utilities interface's two basic methods:

- `write_str()` - writing a command without an answer, for example `*RST`
- `query_str()` - querying your instrument, for example the `*IDN?` query

You may ask a question. Actually, two questions:

- **Q1:** Why there are not called `write()` and `query()` ?
- **Q2:** Where is the `read()` ?

Answer 1: Actually, there are - the `write_str()` / `write()` and `query_str()` / `query()` are aliases, and you can use any of them. We promote the `_str` names, to clearly show you want to work with strings. Strings in Python3 are Unicode, the *bytes* and *string* objects are not interchangeable, since one character might be represented by more than 1 byte. To avoid mixing string and binary communication, all the method names for binary transfer contain `_bin` in the name.

Answer 2: Short answer - you do not need it. Long answer - your instrument never sends unsolicited responses. If you send a set command, you use `write_str()`. For a query command, you use `query_str()`. So, you really do not need it...

Bottom line - if you are used to `write()` and `query()` methods, from pyvisa, the `write_str()` and `query_str()` are their equivalents.

Enough with the theory, let us look at an example. Simple write, and query:

```
"""
Basic string write_str / query_str
"""

from RsCmwCdma2kMeas import *

driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR')
driver.utilities.write_str('*RST')
response = driver.utilities.query_str('*IDN?')
print(response)

# Close the session
driver.close()
```

This example is so-called “*University-Professor-Example*” - good to show a principle, but never used in praxis. The abovementioned commands are already a part of the driver's API. Here is another example, achieving the same goal:

```
"""
Basic string write_str / query_str
"""

from RsCmwCdma2kMeas import *

driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR')
driver.utilities.reset()
print(driver.utilities.idn_string)
```

(continues on next page)

(continued from previous page)

```
# Close the session
driver.close()
```

One additional feature we need to mention here: **VISA timeout**. To simplify, VISA timeout plays a role in each `query_xxx()`, where the controller (your PC) has to prevent waiting forever for an answer from your instrument. VISA timeout defines that maximum waiting time. You can set/read it with the `visa_timeout` property:

```
# Timeout in milliseconds
driver.utilities.visa_timeout = 3000
```

After this time, the `RsCmwCdma2kMeas` raises an exception. Speaking of exceptions, an important feature of the `RsCmwCdma2kMeas` is **Instrument Status Checking**. Check out the next chapter that describes the error checking in details.

For completion, we mention other string-based `write_xxx()` and `query_xxx()` methods - all in one example. They are convenient extensions providing type-safe float/boolean/integer setting/querying features:

```
"""
Basic string write_xxx / query_xxx
"""

from RsCmwCdma2kMeas import *

driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR')
driver.utilities.visa_timeout = 5000
driver.utilities.instrument_status_checking = True
driver.utilities.write_int('SWEEP:COUNT ', 10) # sending 'SWEEP:COUNT 10'
driver.utilities.write_bool('SOURCE:RF:OUTPUT:STATE ', True) # sending
↳ 'SOURCE:RF:OUTPUT:STATE ON'
driver.utilities.write_float('SOURCE:RF:FREQUENCY ', 1E9) # sending 'SOURCE:RF:FREQUENCY_
↳ 10000000000'

sc = driver.utilities.query_int('SWEEP:COUNT?') # returning integer number sc=10
out = driver.utilities.query_bool('SOURCE:RF:OUTPUT:STATE?') # returning boolean_
↳ out=True
freq = driver.utilities.query_float('SOURCE:RF:FREQUENCY?') # returning float number_
↳ freq=1E9

# Close the session
driver.close()
```

Lastly, a method providing basic synchronization: `query_opc()`. It sends query ***OPC?** to your instrument. The instrument waits with the answer until all the tasks it currently has in a queue are finished. This way your program waits too, and this way it is synchronized with the actions in the instrument. Remember to have the VISA timeout set to an appropriate value to prevent the timeout exception. Here's the snippet:

```
driver.utilities.visa_timeout = 3000
driver.utilities.write_str("INIT")
driver.utilities.query_opc()

# The results are ready now to fetch
results = driver.utilities.query_str("FETCH:MEASUREMENT?")
```

Tip: Wait, there's more: you can send the ***OPC?** after each `write_xxx()` automatically:

```
# Default value after init is False
driver.utilities.opc_query_after_write = True
```

1.6 Error Checking

RsCmwCdma2kMeas pushes limits even further (internal R&S joke): It has a built-in mechanism that after each command/query checks the instrument's status subsystem, and raises an exception if it detects an error. For those who are already screaming: **Speed Performance Penalty!!!**, don't worry, you can disable it.

Instrument status checking is very useful since in case your command/query caused an error, you are immediately informed about it. Status checking has in most cases no practical effect on the speed performance of your program. However, if for example, you do many repetitions of short write/query sequences, it might make a difference to switch it off:

```
# Default value after init is True
driver.utilities.instrument_status_checking = False
```

To clear the instrument status subsystem of all errors, call this method:

```
driver.utilities.clear_status()
```

Instrument's status system error queue is clear-on-read. It means, if you query its content, you clear it at the same time. To query and clear list of all the current errors, use this snippet:

```
errors_list = driver.utilities.query_all_errors()
```

See the next chapter on how to react on errors.

1.7 Exception Handling

The base class for all the exceptions raised by the RsCmwCdma2kMeas is `RsInstrException`. Inherited exception classes:

- `ResourceError` raised in the constructor by problems with initiating the instrument, for example wrong or non-existing resource name
- `StatusException` raised if a command or a query generated error in the instrument's error queue
- `TimeoutException` raised if a visa timeout or an opc timeout is reached

In this example we show usage of all of them. Because it is difficult to generate an error using the instrument-specific SCPI API, we use plain SCPI commands:

```
"""
Showing how to deal with exceptions
"""

from RsCmwCdma2kMeas import *
```

(continues on next page)

(continued from previous page)

```

driver = None
# Try-catch for initialization. If an error occurs, the ResourceError is raised
try:
    driver = RsCmwCdma2kMeas('TCPIP::10.112.1.179::HISLIP')
except ResourceError as e:
    print(e.args[0])
    print('Your instrument is probably OFF...')
    # Exit now, no point of continuing
    exit(1)

# Dealing with commands that potentially generate errors OPTION 1:
# Switching the status checking OFF temporarily
driver.utilities.instrument_status_checking = False
driver.utilities.write_str('MY:MISSpelled:COMMAND')
# Clear the error queue
driver.utilities.clear_status()
# Status checking ON again
driver.utilities.instrument_status_checking = True

# Dealing with queries that potentially generate errors OPTION 2:
try:
    # You might want to reduce the VISA timeout to avoid long waiting
    driver.utilities.visa_timeout = 1000
    driver.utilities.query_str('MY:WRONG:QUERy?')

except StatusException as e:
    # Instrument status error
    print(e.args[0])
    print('Nothing to see here, moving on...')

except TimeoutException as e:
    # Timeout error
    print(e.args[0])
    print('That took a long time...')

except RsInstrException as e:
    # RsInstrException is a base class for all the RsCmwCdma2kMeas exceptions
    print(e.args[0])
    print('Some other RsCmwCdma2kMeas error...')

finally:
    driver.utilities.visa_timeout = 5000
    # Close the session in any case
    driver.close()

```

Tip: General rules for exception handling:

- If you are sending commands that might generate errors in the instrument, for example deleting a file which does not exist, use the **OPTION 1** - temporarily disable status checking, send the command, clear the error queue and enable the status checking again.
- If you are sending queries that might generate errors or timeouts, for example querying measurement that can not be performed at the moment, use the **OPTION 2** - try/except with optionally adjusting the timeouts.

1.8 Transferring Files

Instrument -> PC

You definitely experienced it: you just did a perfect measurement, saved the results as a screenshot to an instrument's storage drive. Now you want to transfer it to your PC. With RsCmwCdma2kMeas, no problem, just figure out where the screenshot was stored on the instrument. In our case, it is *var/user/instr_screenshot.png*:

```
driver.utilities.read_file_from_instrument_to_pc(  
    r'var/user/instr_screenshot.png',  
    r'c:\temp\pc_screenshot.png')
```

PC -> Instrument

Another common scenario: Your cool test program contains a setup file you want to transfer to your instrument: Here is the RsCmwCdma2kMeas one-liner split into 3 lines:

```
driver.utilities.send_file_from_pc_to_instrument(  
    r'c:\MyCoolTestProgram\instr_setup.sav',  
    r'var/appdata/instr_setup.sav')
```

1.9 Writing Binary Data

Writing from bytes

An example where you need to send binary data is a waveform file of a vector signal generator. First, you compose your *wform_data* as bytes, and then you send it with *write_bin_block()*:

```
# MyWaveform.wv is an instrument file name under which this data is stored  
driver.utilities.write_bin_block(  
    "SOUR:BB:ARB:WAV:DATA 'MyWaveform.wv'",",  
    wform_data)
```

Note: Notice the *write_bin_block()* has two parameters:

- string parameter *cmd* for the SCPI command
 - bytes parameter *payload* for the actual binary data to send
-

Writing from PC files

Similar to querying binary data to a file, you can write binary data from a file. The second parameter is then the PC file path the content of which you want to send:

```
driver.utilities.write_bin_block_from_file(
    "SOUR:BB:ARB:WAV:DATA 'MyWaveform.wv'",
    r"c:\temp\wform_data.wv")
```

1.10 Transferring Big Data with Progress

We can agree that it can be annoying using an application that shows no progress for long-lasting operations. The same is true for remote-control programs. Luckily, the RsCmwCdma2kMeas has this covered. And, this feature is quite universal - not just for big files transfer, but for any data in both directions.

RsCmwCdma2kMeas allows you to register a function (programmers fancy name is `callback`), which is then periodically invoked after transfer of one data chunk. You can define that chunk size, which gives you control over the callback invoke frequency. You can even slow down the transfer speed, if you want to process the data as they arrive (direction instrument -> PC).

To show this in praxis, we are going to use another *University-Professor-Example*: querying the `*IDN?` with chunk size of 2 bytes and delay of 200ms between each chunk read:

```
"""
Event handlers by reading
"""

from RsCmwCdma2kMeas import *
import time

def my_transfer_handler(args):
    """Function called each time a chunk of data is transferred"""
    # Total size is not always known at the beginning of the transfer
    total_size = args.total_size if args.total_size is not None else "unknown"

    print(f"Context: '{args.context}{'with opc' if args.opc_sync else ''}', "
          f"chunk {args.chunk_ix}, "
          f"transferred {args.transferred_size} bytes, "
          f"total size {total_size}, "
          f"direction {'reading' if args.reading else 'writing'}, "
          f"data '{args.data}'")

    if args.end_of_transfer:
        print('End of Transfer')
        time.sleep(0.2)

driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR')

driver.events.on_read_handler = my_transfer_handler
# Switch on the data to be included in the event arguments
```

(continues on next page)

(continued from previous page)

```
# The event arguments args.data will be updated
driver.events.io_events_include_data = True
# Set data chunk size to 2 bytes
driver.utilities.data_chunk_size = 2
driver.utilities.query_str('*IDN?')
# Unregister the event handler
driver.utilities.on_read_handler = None

# Close the session
driver.close()
```

If you start it, you might wonder (or maybe not): why is the `args.total_size = None`? The reason is, in this particular case the `RsCmwCdma2kMeas` does not know the size of the complete response up-front. However, if you use the same mechanism for transfer of a known data size (for example, file transfer), you get the information about the total size too, and hence you can calculate the progress as:

$$\text{progress [pct]} = 100 * \text{args.transferred_size} / \text{args.total_size}$$

Snippet of transferring file from PC to instrument, the rest of the code is the same as in the previous example:

```
driver.events.on_write_handler = my_transfer_handler
driver.events.io_events_include_data = True
driver.data_chunk_size = 1000
driver.utilities.send_file_from_pc_to_instrument(
    r'c:\MyCoolTestProgram\my_big_file.bin',
    r'var/user/my_big_file.bin')
# Unregister the event handler
driver.events.on_write_handler = None
```

1.11 Multithreading

You are at the party, many people talking over each other. Not every person can deal with such crosstalk, neither can measurement instruments. For this reason, `RsCmwCdma2kMeas` has a feature of scheduling the access to your instrument by using so-called **Locks**. Locks make sure that there can be just one client at a time *talking* to your instrument. Talking in this context means completing one communication step - one command write or write/read or write/read/error check.

To describe how it works, and where it matters, we take three typical multithread scenarios:

One instrument session, accessed from multiple threads

You are all set - the lock is a part of your instrument session. Check out the following example - it will execute properly, although the instrument gets 10 queries at the same time:

```
"""
Multiple threads are accessing one RsCmwCdma2kMeas object
"""

import threading
from RsCmwCdma2kMeas import *
```

(continues on next page)

(continued from previous page)

```

def execute(session):
    """Executed in a separate thread."""
    session.utilities.query_str('*IDN?')

driver = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR')
threads = []
for i in range(10):
    t = threading.Thread(target=execute, args=(driver, ))
    t.start()
    threads.append(t)
print('All threads started')

# Wait for all threads to join this main thread
for t in threads:
    t.join()
print('All threads ended')

driver.close()

```

Shared instrument session, accessed from multiple threads

Same as the previous case, you are all set. The session carries the lock with it. You have two objects, talking to the same instrument from multiple threads. Since the instrument session is shared, the same lock applies to both objects causing the exclusive access to the instrument.

Try the following example:

```

"""
Multiple threads are accessing two RsCmwCdma2kMeas objects with shared session
"""

import threading
from RsCmwCdma2kMeas import *

def execute(session: RsCmwCdma2kMeas, session_ix, index) -> None:
    """Executed in a separate thread."""
    print(f'{index} session {session_ix} query start...')
    session.utilities.query_str('*IDN?')
    print(f'{index} session {session_ix} query end')

driver1 = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR')
driver2 = RsCmwCdma2kMeas.from_existing_session(driver1)
driver1.utilities.visa_timeout = 200
driver2.utilities.visa_timeout = 200
# To see the effect of crosstalk, uncomment this line
# driver2.utilities.clear_lock()

```

(continues on next page)

(continued from previous page)

```

threads = []
for i in range(10):
    t = threading.Thread(target=execute, args=(driver1, 1, i,))
    t.start()
    threads.append(t)
    t = threading.Thread(target=execute, args=(driver2, 2, i,))
    t.start()
    threads.append(t)
print('All threads started')

# Wait for all threads to join this main thread
for t in threads:
    t.join()
print('All threads ended')

driver2.close()
driver1.close()

```

As you see, everything works fine. If you want to simulate some party crosstalk, uncomment the line `driver2.utilities.clear_lock()`. This causes the driver2 session lock to break away from the driver1 session lock. Although the driver1 still tries to schedule its instrument access, the driver2 tries to do the same at the same time, which leads to all the fun stuff happening.

Multiple instrument sessions accessed from multiple threads

Here, there are two possible scenarios depending on the instrument's VISA interface:

- You are lucky, because your instrument handles each remote session completely separately. An example of such instrument is SMW200A. In this case, you have no need for session locking.
- Your instrument handles all sessions with one set of in/out buffers. You need to lock the session for the duration of a talk. And you are lucky again, because the RsCmwCdma2kMeas takes care of it for you. The text below describes this scenario.

Run the following example:

```

"""
Multiple threads are accessing two RsCmwCdma2kMeas objects with two separate sessions
"""

import threading
from RsCmwCdma2kMeas import *

def execute(session: RsCmwCdma2kMeas, session_ix, index) -> None:
    """Executed in a separate thread."""
    print(f'{index} session {session_ix} query start...')
    session.utilities.query_str('*IDN?')
    print(f'{index} session {session_ix} query end')

driver1 = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR')
driver2 = RsCmwCdma2kMeas('TCPIP::192.168.56.101::INSTR')

```

(continues on next page)

(continued from previous page)

```
driver1.utilities.visa_timeout = 200
driver2.utilities.visa_timeout = 200

# Synchronise the sessions by sharing the same lock
driver2.utilities.assign_lock(driver1.utilities.get_lock()) # To see the effect of
↳ crosstalk, comment this line

threads = []
for i in range(10):
    t = threading.Thread(target=execute, args=(driver1, 1, i,))
    t.start()
    threads.append(t)
    t = threading.Thread(target=execute, args=(driver2, 2, i,))
    t.start()
    threads.append(t)
print('All threads started')

# Wait for all threads to join this main thread
for t in threads:
    t.join()
print('All threads ended')

driver2.close()
driver1.close()
```

You have two completely independent sessions that want to talk to the same instrument at the same time. This will not go well, unless they share the same session lock. The key command to achieve this is `driver2.utilities.assign_lock(driver1.utilities.get_lock())`. Try to comment it and see how it goes. If despite commenting the line the example runs without issues, you are lucky to have an instrument similar to the SMW200A.

REVISION HISTORY

Rohde & Schwarz CMW Base System RsCmwBase instrument driver.

Supported instruments: CMW500, CMW100, CMW270, CMW280

The package is hosted here: <https://pypi.org/project/RsCmwBase/>

Documentation: <https://RsCmwBase.readthedocs.io/>

Examples: <https://github.com/Rohde-Schwarz/Examples/>

Currently supported CMW subsystems:

- Base: RsCmwBase
- Global Purpose RF: RsCmwGprfGen, RsCmwGprfMeas
- Bluetooth: RsCmwBluetoothSig, RsCmwBluetoothMeas
- LTE: RsCmwLteSig, RsCmwLteMeas
- CDMA2000: RsCdma2kSig, RsCdma2kMeas
- 1xEVDO: RsCmwEvdoSig, RsCmwEvdoMeas
- WCDMA: RsCmwWcdmaSig, RsCmwWcdmaMeas
- GSM: RsCmwGsmSig, RsCmwGsmMeas
- WLAN: RsCmwWlanSig, RsCmwWlanMeas
- DAU: RsCmwDau

In case you require support for more subsystems, please contact our customer support on customersupport@rohde-schwarz.com with the topic “Auto-generated Python drivers” in the email subject. This will speed up the response process

Examples: Download the file ‘CMW Python instrument drivers’ from https://www.rohde-schwarz.com/driver/cmw500_overview/ The zip file contains the examples on how to use these drivers. Remember to adjust the resource-Name string to fit your instrument.

Release Notes for the whole RsCmwXXX group:

Latest release notes summary: <INVALID>

Version 3.7.90.39

- <INVALID>
-

Version 3.8.xx2

- Fixed several misspelled arguments and command headers

Version 3.8.xx1

- Bluetooth and WLAN update for FW versions 3.8.xxx

Version 3.7.xx8

- Added documentation on ReadTheDocs

Version 3.7.xx7

- Added 3G measurement subsystems RsCmwGsmMeas, RsCmwCdma2kMeas, RsCmwEvdoMeas, RsCmwWcdmaMeas
- Added new data types for commands accepting numbers or ON/OFF:
 - int or bool
 - float or bool

Version 3.7.xx6

- Added new UDF integer number recognition

Version 3.7.xx5

- Added RsCmwDau

Version 3.7.xx4

- Fixed several interface names
- New release for CMW Base 3.7.90
- New release for CMW Bluetooth 3.7.90

Version 3.7.xx3

- Second release of the CMW python drivers packet
- New core component RsInstrument
- Previously, the groups starting with CATalog: e.g. 'CATalog:SIGNaling:TOPology:PLMN' were reordered to 'SIGNaling:TOPology:PLMN:CATALOG' give more contextual meaning to the method/property name. This is now reverted back, since it was hard to find the desired functionality.
- Reorganized Utilities interface to sub-groups

Version 3.7.xx2

- Fixed some misspelling errors
- Changed enum and repCap types names
- All the assemblies are signed with Rohde & Schwarz signature

Version 1.0.0.0

- First released version

3.1 BandClass

```
# First value:
value = enums.BandClass.AWS
# Last value:
value = enums.BandClass.USPC
# All values (22x):
AWS | B18M | IEXT | IM2K | JTAC | KCEL | KPCS | LBAND
LO7C | N45T | NA7C | NA8S | NA9C | NAPC | PA4M | PA8M
PS7C | SBAND | TACS | U25B | USC | USPC
```

3.2 CmwsConnector

```
# First value:
value = enums.CmwsConnector.R11
# Last value:
value = enums.CmwsConnector.RB8
# All values (48x):
R11 | R12 | R13 | R14 | R15 | R16 | R17 | R18
R21 | R22 | R23 | R24 | R25 | R26 | R27 | R28
R31 | R32 | R33 | R34 | R35 | R36 | R37 | R38
R41 | R42 | R43 | R44 | R45 | R46 | R47 | R48
RA1 | RA2 | RA3 | RA4 | RA5 | RA6 | RA7 | RA8
RB1 | RB2 | RB3 | RB4 | RB5 | RB6 | RB7 | RB8
```

3.3 ParameterSetMode

```
# Example value:
value = enums.ParameterSetMode.GLOBal
# All values (2x):
GLOBal | LIST
```

3.4 Rbw

```
# First value:  
value = enums.Rbw.F100k  
# Last value:  
value = enums.Rbw.F6K25  
# All values (10x):  
F100k | F10K | F12K5 | F1K0 | F1M0 | F1M23 | F25K | F30K  
F50K | F6K25
```

3.5 Rconfig

```
# Example value:  
value = enums.Rconfig.R12Q  
# All values (3x):  
R12Q | R36H | R3Q
```

3.6 Repeat

```
# Example value:  
value = enums.Repeat.CONTinuous  
# All values (2x):  
CONTinuous | SINGleshot
```

3.7 ResourceState

```
# Example value:  
value = enums.ResourceState.ACTive  
# All values (8x):  
ACTive | ADJusted | INValid | OFF | PENDing | QUEued | RDY | RUN
```

3.8 ResultStatus2

```
# First value:  
value = enums.ResultStatus2.DC  
# Last value:  
value = enums.ResultStatus2.ULEU  
# All values (10x):  
DC | INV | NAV | NCAP | OFF | OFL | OK | UFL  
ULEL | ULEU
```

3.9 RetriggerMode

```
# Example value:
value = enums.RettriggerMode.ONCE
# All values (2x):
ONCE | SEGMENT
```

3.10 RetriggerOption

```
# Example value:
value = enums.RettriggerOption.IFPower
# All values (4x):
IFPower | IFPSync | OFF | ON
```

3.11 RxConnector

```
# First value:
value = enums.RxConnector.I11I
# Last value:
value = enums.RxConnector.RH8
# All values (154x):
I11I | I13I | I15I | I17I | I21I | I23I | I25I | I27I
I31I | I33I | I35I | I37I | I41I | I43I | I45I | I47I
IF1 | IF2 | IF3 | IQ1I | IQ3I | IQ5I | IQ7I | R11
R11C | R12 | R12C | R12I | R13 | R13C | R14 | R14C
R14I | R15 | R16 | R17 | R18 | R21 | R21C | R22
R22C | R22I | R23 | R23C | R24 | R24C | R24I | R25
R26 | R27 | R28 | R31 | R31C | R32 | R32C | R32I
R33 | R33C | R34 | R34C | R34I | R35 | R36 | R37
R38 | R41 | R41C | R42 | R42C | R42I | R43 | R43C
R44 | R44C | R44I | R45 | R46 | R47 | R48 | RA1
RA2 | RA3 | RA4 | RA5 | RA6 | RA7 | RA8 | RB1
RB2 | RB3 | RB4 | RB5 | RB6 | RB7 | RB8 | RC1
RC2 | RC3 | RC4 | RC5 | RC6 | RC7 | RC8 | RD1
RD2 | RD3 | RD4 | RD5 | RD6 | RD7 | RD8 | RE1
RE2 | RE3 | RE4 | RE5 | RE6 | RE7 | RE8 | RF1
RF1C | RF2 | RF2C | RF2I | RF3 | RF3C | RF4 | RF4C
RF4I | RF5 | RF5C | RF6 | RF6C | RF7 | RF8 | RFAC
RFBC | RFBI | RG1 | RG2 | RG3 | RG4 | RG5 | RG6
RG7 | RG8 | RH1 | RH2 | RH3 | RH4 | RH5 | RH6
RH7 | RH8
```

3.12 RxConverter

```
# First value:
value = enums.RxConverter.IRX1
# Last value:
value = enums.RxConverter.RX44
# All values (40x):
IRX1 | IRX11 | IRX12 | IRX13 | IRX14 | IRX2 | IRX21 | IRX22
IRX23 | IRX24 | IRX3 | IRX31 | IRX32 | IRX33 | IRX34 | IRX4
IRX41 | IRX42 | IRX43 | IRX44 | RX1 | RX11 | RX12 | RX13
RX14 | RX2 | RX21 | RX22 | RX23 | RX24 | RX3 | RX31
RX32 | RX33 | RX34 | RX4 | RX41 | RX42 | RX43 | RX44
```

3.13 Scenario

```
# Example value:
value = enums.Scenario.CSPath
# All values (4x):
CSPath | MAPRotocol | SALone | UNDEFINED
```

3.14 SigChStateA

```
# Example value:
value = enums.SigChStateA.Active
# All values (4x):
Active | ALIased | IActive | INVisible
```

3.15 SigChStateB

```
# Example value:
value = enums.SigChStateB.Active
# All values (3x):
Active | IActive | INVisible
```

3.16 Slope

```
# Example value:
value = enums.Slope.FEDGE
# All values (4x):
FEDGE | OFF | ON | REDGE
```

3.17 SpreadingFactor

```
# Example value:  
value = enums.SpreadingFactor.SF16  
# All values (3x):  
SF16 | SF32 | SF64
```

3.18 StatePower

```
# Example value:  
value = enums.StatePower.BOTH  
# All values (4x):  
BOTH | LOWer | OK | UPPer
```

3.19 StopCondition

```
# Example value:  
value = enums.StopCondition.NONE  
# All values (2x):  
NONE | SLFail
```

3.20 Tab

```
# Example value:  
value = enums.Tab.MEVA  
# All values (2x):  
MEVA | OLTR
```

3.21 UpDownDirection

```
# Example value:  
value = enums.UpDownDirection.DOWN  
# All values (2x):  
DOWN | UP
```


REPCAPS

4.1 Instance (Global)

```
# Setting:
driver.repcap_instance_set(repcap.Instance.Inst1)
# Range:
Inst1 .. Inst16
# All values (16x):
Inst1 | Inst2 | Inst3 | Inst4 | Inst5 | Inst6 | Inst7 | Inst8
Inst9 | Inst10 | Inst11 | Inst12 | Inst13 | Inst14 | Inst15 | Inst16
```

4.2 AcpMinus

```
# First value:
value = repcap.AcpMinus.Ch1
# Range:
Ch1 .. Ch20
# All values (20x):
Ch1 | Ch2 | Ch3 | Ch4 | Ch5 | Ch6 | Ch7 | Ch8
Ch9 | Ch10 | Ch11 | Ch12 | Ch13 | Ch14 | Ch15 | Ch16
Ch17 | Ch18 | Ch19 | Ch20
```

4.3 AcpPlus

```
# First value:
value = repcap.AcpPlus.Ch1
# Range:
Ch1 .. Ch20
# All values (20x):
Ch1 | Ch2 | Ch3 | Ch4 | Ch5 | Ch6 | Ch7 | Ch8
Ch9 | Ch10 | Ch11 | Ch12 | Ch13 | Ch14 | Ch15 | Ch16
Ch17 | Ch18 | Ch19 | Ch20
```

4.4 Segment

```
# First value:
value = repcap.Segment.Nr1
# Range:
Nr1 .. Nr200
# All values (200x):
Nr1 | Nr2 | Nr3 | Nr4 | Nr5 | Nr6 | Nr7 | Nr8
Nr9 | Nr10 | Nr11 | Nr12 | Nr13 | Nr14 | Nr15 | Nr16
Nr17 | Nr18 | Nr19 | Nr20 | Nr21 | Nr22 | Nr23 | Nr24
Nr25 | Nr26 | Nr27 | Nr28 | Nr29 | Nr30 | Nr31 | Nr32
Nr33 | Nr34 | Nr35 | Nr36 | Nr37 | Nr38 | Nr39 | Nr40
Nr41 | Nr42 | Nr43 | Nr44 | Nr45 | Nr46 | Nr47 | Nr48
Nr49 | Nr50 | Nr51 | Nr52 | Nr53 | Nr54 | Nr55 | Nr56
Nr57 | Nr58 | Nr59 | Nr60 | Nr61 | Nr62 | Nr63 | Nr64
Nr65 | Nr66 | Nr67 | Nr68 | Nr69 | Nr70 | Nr71 | Nr72
Nr73 | Nr74 | Nr75 | Nr76 | Nr77 | Nr78 | Nr79 | Nr80
Nr81 | Nr82 | Nr83 | Nr84 | Nr85 | Nr86 | Nr87 | Nr88
Nr89 | Nr90 | Nr91 | Nr92 | Nr93 | Nr94 | Nr95 | Nr96
Nr97 | Nr98 | Nr99 | Nr100 | Nr101 | Nr102 | Nr103 | Nr104
Nr105 | Nr106 | Nr107 | Nr108 | Nr109 | Nr110 | Nr111 | Nr112
Nr113 | Nr114 | Nr115 | Nr116 | Nr117 | Nr118 | Nr119 | Nr120
Nr121 | Nr122 | Nr123 | Nr124 | Nr125 | Nr126 | Nr127 | Nr128
Nr129 | Nr130 | Nr131 | Nr132 | Nr133 | Nr134 | Nr135 | Nr136
Nr137 | Nr138 | Nr139 | Nr140 | Nr141 | Nr142 | Nr143 | Nr144
Nr145 | Nr146 | Nr147 | Nr148 | Nr149 | Nr150 | Nr151 | Nr152
Nr153 | Nr154 | Nr155 | Nr156 | Nr157 | Nr158 | Nr159 | Nr160
Nr161 | Nr162 | Nr163 | Nr164 | Nr165 | Nr166 | Nr167 | Nr168
Nr169 | Nr170 | Nr171 | Nr172 | Nr173 | Nr174 | Nr175 | Nr176
Nr177 | Nr178 | Nr179 | Nr180 | Nr181 | Nr182 | Nr183 | Nr184
Nr185 | Nr186 | Nr187 | Nr188 | Nr189 | Nr190 | Nr191 | Nr192
Nr193 | Nr194 | Nr195 | Nr196 | Nr197 | Nr198 | Nr199 | Nr200
```

4.5 Sequence

```
# First value:
value = repcap.Sequence.Nr1
# Range:
Nr1 .. Nr5
# All values (5x):
Nr1 | Nr2 | Nr3 | Nr4 | Nr5
```

EXAMPLES

For more examples, visit our [Rohde & Schwarz Github repository](#).

```
""" Example on how to use the python RsCmw auto-generated instrument driver showing:
- usage of basic properties of the cmw_base object
- basic concept of setting commands and repcaps: DISPLAY:WINDOW<n>:SElect
- cmw_xxx drivers reliability interface usage
"""

from RsCmwBase import * # install from pypi.org

RsCmwBase.assert_minimum_version('3.7.90.32')
cmw_base = RsCmwBase('TCPIP::10.112.1.116::INSTR', True, False)
print(f'CMW Base IND: {cmw_base.utilities.idn_string}')
print(f'CMW Instrument options:\n{"", ".join(cmw_base.utilities.instrument_options)}')
cmw_base.utilities.visa_timeout = 5000

# Sends OPC after each command
cmw_base.utilities.opc_query_after_write = False

# Checks for syst:err? after each command / query
cmw_base.utilities.instrument_status_checking = True

# DISPLAY:WINDOW<n>:SElect
cmw_base.display.window.select.set(repcap.Window.Win1)
cmw_base.display.window.repcap_window_set(repcap.Window.Win2)
cmw_base.display.window.select.set()

# Self-test
self_test = cmw_base.utilities.self_test()
print(f'CMW self-test result: {self_test} - {"Passed" if self_test[0] == 0 else "Failed"}')
↪ ''

# Driver's Interface reliability offers a convenient way of reacting on the return value.
↪ Reliability Indicator
cmw_base.reliability.ExceptionOnError = True

# Callback to use for the reliability indicator update event
def my_reliability_handler(event_args: ReliabilityEventArgs):
    print(f'Base Reliability updated.\nContext: {event_args.context}\nMessage:
↪ {event_args.message}')
```

(continues on next page)

(continued from previous page)

```
# We register a callback for each change in the reliability indicator
cmw_base.reliability.on_update_handler = my_reliability_handler

# You can obtain the last value of the returned reliability
print(f"\nReliability last value: {cmw_base.reliability.last_value}, context '{cmw_base.
↳reliability.last_context}', message: {cmw_base.reliability.last_message}")

# Reference Frequency Source
cmw_base.system.reference.frequency.source_set(enums.SourceIntExt.INTERNAL)

# Close the session
cmw_base.close()
```

**CHAPTER
SIX**

INDEX

RSCMWCDMA2KMEAS API STRUCTURE

Global RepCaps

```
driver = RsCmwCdma2kMeas('TCPIP::192.168.2.101::HISLIP')
# Instance range: Inst1 .. Inst16
rc = driver.repcap_instance_get()
driver.repcap_instance_set(repcap.Instance.Inst1)
```

class RsCmwCdma2kMeas(resource_name: str, id_query: bool = True, reset: bool = False, options: Optional[str] = None, direct_session: Optional[object] = None)

777 total commands, 6 Sub-groups, 0 group commands

Initializes new RsCmwCdma2kMeas session.

Parameter options tokens examples:

- 'Simulate=True' - starts the session in simulation mode. Default: False
- 'SelectVisa=socket' - uses no VISA implementation for socket connections - you do not need any VISA-C installation
- 'SelectVisa=rs' - forces usage of RohdeSchwarz Visa
- 'SelectVisa=ni' - forces usage of National Instruments Visa
- 'QueryInstrumentStatus = False' - same as driver.utilities.instrument_status_checking = False
- 'DriverSetup=(WriteDelay = 20, ReadDelay = 5)' - Introduces delay of 20ms before each write and 5ms before each read
- 'DriverSetup=(OpcWaitMode = OpcQuery)' - mode for all the opc-synchronised write/reads. Other modes: StbPolling, StbPollingSlow, StbPollingSuperSlow
- 'DriverSetup=(AddTermCharToWriteBinBLock = True)' - Adds one additional LF to the end of the binary data (some instruments require that)
- 'DriverSetup=(AssureWriteWithTermChar = True)' - Makes sure each command/query is terminated with termination character. Default: Interface dependent
- 'DriverSetup=(TerminationCharacter = 'x')' - Sets the termination character for reading. Default: '<LF>' (LineFeed)
- 'DriverSetup=(IoSegmentSize = 10E3)' - Maximum size of one write/read segment. If transferred data is bigger, it is split to more segments
- 'DriverSetup=(OpcTimeout = 10000)' - same as driver.utilities.opc_timeout = 10000
- 'DriverSetup=(VisaTimeout = 5000)' - same as driver.utilities.visa_timeout = 5000

- ‘DriverSetup=(ViClearExeMode = 255)’ - Binary combination where 1 means performing viClear() on a certain interface as the very first command in init
- ‘DriverSetup=(OpcQueryAfterWrite = True)’ - same as driver.utilities.opc_query_after_write = True

Parameters

- **resource_name** – VISA resource name, e.g. ‘TCPIP::192.168.2.1::INSTR’
- **id_query** – if True: the instrument’s model name is verified against the models supported by the driver and eventually throws an exception.
- **reset** – Resets the instrument (sends *RST command) and clears its status sybsystem
- **options** – string tokens alternating the driver settings.
- **direct_session** – Another driver object or pyVisa object to reuse the session instead of opening a new session.

static assert_minimum_version(min_version: str) → None

Asserts that the driver version fulfills the minimum required version you have entered. This way you make sure your installed driver is of the entered version or newer.

close() → None

Closes the active RsCmwCdma2kMeas session.

classmethod from_existing_session(session: object, options: Optional[str] = None) → RsCmwCdma2kMeas

Creates a new RsCmwCdma2kMeas object with the entered ‘session’ reused.

Parameters

- **session** – can be an another driver or a direct pyvisa session.
- **options** – string tokens alternating the driver settings.

get_session_handle() → object

Returns the underlying session handle.

static list_resources(expression: str = '?*::INSTR', visa_select: Optional[str] = None) → List[str]

Finds all the resources defined by the expression

- ‘?*’ - matches all the available instruments
- ‘USB::?*’ - matches all the USB instruments
- ‘TCPIP::192?*’ - matches all the LAN instruments with the IP address starting with 192

Parameters

- **expression** – see the examples in the function
- **visa_select** – optional parameter selecting a specific VISA. Examples: ‘@ni’, ‘@rs’

restore_all_repcaps_to_default() → None

Sets all the Group and Global repcaps to their initial values

Subgroups

7.1 Route

SCPI Commands

```
ROUTE:CDMA:MEASurement<Instance>
```

class Route

Route commands group definition. 5 total commands, 1 Sub-groups, 1 group commands

class ValueStruct

Structure for reading output parameters. Fields:

- Scenario: enums.Scenario: SALone | CSPath SALone: Standalone (non-signaling) CPath: Combined signal path
- Controller: str: string Controlling application for scenario CPath
- Rx_Connector: enums.RxConnector: RF connector for the input path
- Rx_Converter: enums.RxConverter: RX module for the input path

get_value() → ValueStruct

```
# SCPI: ROUTE:CDMA:MEASurement<Instance>
value: ValueStruct = driver.route.get_value()
```

Returns the configured routing settings. For possible connector and converter values, see ‘Values for RF Path Selection’.

return structure: for return value, see the help for ValueStruct structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.route.clone()
```

Subgroups

7.1.1 Scenario

SCPI Commands

```
ROUTE:CDMA:MEASurement<Instance>:SCENario:SALone
ROUTE:CDMA:MEASurement<Instance>:SCENario:CSPath
ROUTE:CDMA:MEASurement<Instance>:SCENario
```

class Scenario

Scenario commands group definition. 4 total commands, 1 Sub-groups, 3 group commands

class SaloneStruct

Structure for reading output parameters. Fields:

- Rx_Connector: enums.RxConnector: RF connector for the input path
- Rf_Converter: enums.RxConverter: RX module for the input path

get_cspath() → str

```
# SCPI: ROUTe:CDMA:MEASurement<Instance>:SCENario:CSPath
value: str = driver.route.scenario.get_cspath()
```

Activates the combined signal path scenario and selects a master. The master controls the signal routing and analyzer settings while the combined signal path scenario is active. Depending on the installed options, the set of masters available at your instrument can differ from the values listed below. A complete list of all supported values can be displayed using method RsCmwCdma2kMeas.Route.Scenario.Catalog.cspath.

return master: string String parameter containing the master application, e.g. 'CDMA2000 Sig1' or 'CDMA2000 Sig2'

get_salone() → SaloneStruct

```
# SCPI: ROUTe:CDMA:MEASurement<Instance>:SCENario:SALone
value: SaloneStruct = driver.route.scenario.get_salone()
```

Activates the standalone scenario and selects the RF input path for the measured RF signal. For possible connector and converter values, see 'Values for RF Path Selection'.

return structure: for return value, see the help for SaloneStruct structure arguments.

get_value() → RsCmwCdma2kMeas.enums.Scenario

```
# SCPI: ROUTe:CDMA:MEASurement<Instance>:SCENario
value: enums.Scenario = driver.route.scenario.get_value()
```

Returns the active scenario.

return scenario: SALone | CSPath SALone: Standalone (non-signaling) CSPath: Combined signal path

set_cspath(master: str) → None

```
# SCPI: ROUTe:CDMA:MEASurement<Instance>:SCENario:CSPath
driver.route.scenario.set_cspath(master = '1')
```

Activates the combined signal path scenario and selects a master. The master controls the signal routing and analyzer settings while the combined signal path scenario is active. Depending on the installed options, the set of masters available at your instrument can differ from the values listed below. A complete list of all supported values can be displayed using method RsCmwCdma2kMeas.Route.Scenario.Catalog.cspath.

param master string String parameter containing the master application, e.g. 'CDMA2000 Sig1' or 'CDMA2000 Sig2'

set_salone(value: RsCmwCdma2kMeas.Implementations.Route_Scenario.Scenario.SaloneStruct) → None

```
# SCPI: ROUTe:CDMA:MEASurement<Instance>:SCENario:SALone
driver.route.scenario.set_salone(value = SaloneStruct())
```

Activates the standalone scenario and selects the RF input path for the measured RF signal. For possible connector and converter values, see ‘Values for RF Path Selection’.

param value see the help for SaloneStruct structure arguments.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.route.scenario.clone()
```

Subgroups

7.1.1.1 Catalog

SCPI Commands

```
ROUTE:CDMA:MEASurement<Instance>:SCENario:CATalog:CSPath
```

class Catalog

Catalog commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

get_cspath() → List[str]

```
# SCPI: ROUTe:CDMA:MEASurement<Instance>:SCENario:CATalog:CSPath
value: List[str] = driver.route.scenario.catalog.get_cspath()
```

Lists all applications that can be set as master for the combined signal path scenario using method RsCmwCdma2kMeas.Route.Scenario.cspath.

return source_list: string Comma-separated list. Each supported value is represented as a string.

7.2 Configure

SCPI Commands

```
CONFigure:CDMA:MEASurement<Instance>:RCONfig
CONFigure:CDMA:MEASurement<Instance>:DISPlay
```

class Configure

Configure commands group definition. 70 total commands, 3 Sub-groups, 2 group commands

get_display() → RsCmwCdma2kMeas.enums.Tab

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:DISPlay
value: enums.Tab = driver.configure.get_display()
```

Selects the view to be shown when the display is switched on during remote control.

return tab: MEVA | OLTR Multi-evaluation - overview, OLTR view

get_rconfig() → RsCmwCdma2kMeas.enums.Rconfig

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RCONfig
value: enums.Rconfig = driver.configure.get_rconfig()
```

Selects the radio configuration which determines, for example, the modulation type. IN-TRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:CDMA:SIGN<i>:PREConfigure:LAYer:RCONfig during no active call CONFIGure:CDMA:SIGN<i>:LAYer:RCONfig during call established
- CONFIGure:CDMA:SIGN<i>:LAYer:MODulation

return rconfig: R12Q | R36H | R3Q R12Q: RC1 or 2 (O-QPSK) R36H: RC3 to 6 (H-PSK) R3Q: RC3 (QPSK)

set_display(tab: RsCmwCdma2kMeas.enums.Tab) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:DISPlay
driver.configure.set_display(tab = enums.Tab.MEVA)
```

Selects the view to be shown when the display is switched on during remote control.

param tab MEVA | OLTR Multi-evaluation - overview, OLTR view

set_rconfig(rconfig: RsCmwCdma2kMeas.enums.Rconfig) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RCONfig
driver.configure.set_rconfig(rconfig = enums.Rconfig.R12Q)
```

Selects the radio configuration which determines, for example, the modulation type. IN-TRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:CDMA:SIGN<i>:PREConfigure:LAYer:RCONfig during no active call CONFIGure:CDMA:SIGN<i>:LAYer:RCONfig during call established
- CONFIGure:CDMA:SIGN<i>:LAYer:MODulation

param rconfig R12Q | R36H | R3Q R12Q: RC1 or 2 (O-QPSK) R36H: RC3 to 6 (H-PSK) R3Q: RC3 (QPSK)

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.clone()
```

Subgroups

7.2.1 RfSettings

SCPI Commands

```

CONFigure:CDMA:MEASurement<Instance>:RFSettings:FREquency
CONFigure:CDMA:MEASurement<Instance>:RFSettings:CHANnel
CONFigure:CDMA:MEASurement<Instance>:RFSettings:EATTenuation
CONFigure:CDMA:MEASurement<Instance>:RFSettings:UMARgin
CONFigure:CDMA:MEASurement<Instance>:RFSettings:ENPower
CONFigure:CDMA:MEASurement<Instance>:RFSettings:BClass
CONFigure:CDMA:MEASurement<Instance>:RFSettings:FOFFset

```

class RfSettings

RfSettings commands group definition. 7 total commands, 0 Sub-groups, 7 group commands

get_bclass() → RsCmwCdma2kMeas.enums.BandClass

```

# SCPI: CONFigure:CDMA:MEASurement<Instance>:RFSettings:BClass
value: enums.BandClass = driver.configure.rfSettings.get_bclass()

```

Selects the band class (BC) . If the current center frequency (method RsCmwCdma2kMeas.Configure.RfSettings.frequency) is valid for this band class, the corresponding channel number (method RsCmwCdma2kMeas.Configure.RfSettings.channel) is also calculated and set. See also 'Band Classes'. For the combined signal path scenario, useCONFigure:CDMA:SIGN<i>:RFSettings:BClass.

return band_class: USC | KCEL | NAPC | TACS | JTAC | KPCS | N45T | IM2K | NA7C | B18M | NA8S | PA4M | PA8M | IEXT | USPC | AWS | U25B | NA9C | PS7C | LO7C | LBANd | SBANd
 USC: BC 0, US-Cellular KCEL: BC 0, Korean Cellular NAPC: BC 1, North American PCS TACS: BC 2, TACS Band JTAC: BC 3, JTACS Band KPCS: BC 4, Korean PCS N45T: BC 5, NMT-450 IM2K: BC 6, IMT-2000 NA7C: BC 7, Upper 700 MHz B18M: BC 8, 1800 MHz Band NA9C: BC 9, North American 900 MHz NA8S: BC 10, Secondary 800 MHz PA4M: BC 11, European 400 MHz PAMR PA8M: BC 12, 800 MHz PAMR IEXT: BC 13, IMT-2000 2.5 GHz Extension USPC: BC 14, US PCS 1900 MHz AWS: BC 15, AWS Band U25B: BC 16, US 2.5 GHz Band PS7C: BC 18, Public Safety Band 700 MHz LO7C: BC 19, Lower 700 MHz LBAN: BC 20, L-Band SBAN: BC 21, S-Band

get_channel() → int

```

# SCPI: CONFigure:CDMA:MEASurement<Instance>:RFSettings:CHANnel
value: int = driver.configure.rfSettings.get_channel()

```

Selects the channel number. The channel number must be valid for the current band class, for dependencies see 'Band Classes'. The corresponding center frequency (method RsCmwCdma2kMeas.Configure.RfSettings.frequency) is calculated and set. For the combined signal path scenario, useCONFigure:CDMA:SIGN<i>:RFSettings:CHANnel.

return channel: integer Range: Depends on selected band class

get_eattenuation() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:EATTenuation
value: float = driver.configure.rfSettings.get_eattenuation()
```

Defines an external attenuation (or gain, if the value is negative) , to be applied to the input connector. For the combined signal path scenario, useCONFIGure:CDMA:SIGN<i>:RFSettings:EATTenuation.

return rf_input_ext_att: numeric Range: -50 dB to 90 dB, Unit: dB

get_envelope_power() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:ENPower
value: float = driver.configure.rfSettings.get_envelope_power()
```

Sets the expected nominal power of the measured RF signal. INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:CDMA:SIGN<i>:RFPower:EPMode
- CONFIGure:CDMA:SIGN<i>:RFPower:MANual
- CONFIGure:CDMA:SIGN<i>:RFPower:EXpected

return exp_nominal_power: numeric The range of the expected nominal power can be calculated as follows: Range (Expected Nominal Power) = Range (Input Power) + External Attenuation - User Margin The input power range is stated in the data sheet. Unit: dBm

get_foffset() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:FOFFset
value: float = driver.configure.rfSettings.get_foffset()
```

Selects a positive or negative offset frequency to be added to the center frequency (method RsCmwCdma2kMeas.Configure. RfSettings.frequency) . For the combined signal path scenario, useCONFIGure:CDMA:SIGN<i>:RFSettings:FOFFset.

return frequency_offset: numeric Range: -50 kHz to 50 kHz, Unit: Hz

get_frequency() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:FREquency
value: float = driver.configure.rfSettings.get_frequency()
```

Selects the center frequency of the RF analyzer. If the center frequency is valid for the current band class, the corresponding channel number is also calculated and set.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:CDMA:SIGN<i>:RFSettings:RLFrequency
- CONFIGure:CDMA:SIGN<i>:RFSettings:FREquency or
- CONFIGure:CDMA:SIGN<i>:RFSettings:CHANnel

The supported frequency range depends on the instrument model and the available options. The supported range can be smaller than stated here. Refer to the preface of your model-specific base unit manual.

return frequency: numeric Range: 100 MHz to 6 GHz, Unit: Hz

get_umargin() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:UMARgin
value: float = driver.configure.rfSettings.get_umargin()
```

Sets the margin that the measurement adds to the expected nominal power to determine the reference power. The reference power minus the external input attenuation must be within the power range of the selected input connector. Refer to the data sheet.

return user_margin: numeric Range: 0 dB to (55 dB + External Attenuation - Expected Nominal Power) , Unit: dB

set_bclass(band_class: RsCmwCdma2kMeas.enums.BandClass) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:BCLass
driver.configure.rfSettings.set_bclass(band_class = enums.BandClass.AWS)
```

Selects the band class (BC) . If the current center frequency (method RsCmwCdma2kMeas.Configure.RfSettings.frequency) is valid for this band class, the corresponding channel number (method RsCmwCdma2kMeas.Configure.RfSettings.channel) is also calculated and set. See also 'Band Classes'. For the combined signal path scenario, useCONFIGure:CDMA:SIGN<i>:RFSettings:BCLass.

param band_class USC | KCEL | NAPC | TACS | JTAC | KPCS | N45T | IM2K | NA7C | B18M | NA8S | PA4M | PA8M | IEXT | USPC | AWS | U25B | NA9C | PS7C | LO7C | LBANd | SBANd
USC: BC 0, US-Cellular KCEL: BC 0, Korean Cellular NAPC: BC 1, North American PCS TACS: BC 2, TACS Band JTAC: BC 3, JTACS Band KPCS: BC 4, Korean PCS N45T: BC 5, NMT-450 IM2K: BC 6, IMT-2000 NA7C: BC 7, Upper 700 MHz B18M: BC 8, 1800 MHz Band NA9C: BC 9, North American 900 MHz NA8S: BC 10, Secondary 800 MHz PA4M: BC 11, European 400 MHz PAMR PA8M: BC 12, 800 MHz PAMR IEXT: BC 13, IMT-2000 2.5 GHz Extension USPC: BC 14, US PCS 1900 MHz AWS: BC 15, AWS Band U25B: BC 16, US 2.5 GHz Band PS7C: BC 18, Public Safety Band 700 MHz LO7C: BC 19, Lower 700 MHz LBAN: BC 20, L-Band SBAN: BC 21, S-Band

set_channel(channel: int) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:CHANnel
driver.configure.rfSettings.set_channel(channel = 1)
```

Selects the channel number. The channel number must be valid for the current band class, for dependencies see 'Band Classes'. The corresponding center frequency (method RsCmwCdma2kMeas.Configure.RfSettings.frequency) is calculated and set. For the combined signal path scenario, useCONFIGure:CDMA:SIGN<i>:RFSettings:CHANnel.

param channel integer Range: Depends on selected band class

set_eattenuation(rf_input_ext_att: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:EATTenuation
driver.configure.rfSettings.set_eattenuation(rf_input_ext_att = 1.0)
```

Defines an external attenuation (or gain, if the value is negative) , to be applied to the input connector. For the combined signal path scenario, useCONFIGure:CDMA:SIGN<i>:RFSettings:EATTenuation.

param rf_input_ext_att numeric Range: -50 dB to 90 dB, Unit: dB

set_envelope_power(*exp_nominal_power: float*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:ENPower
driver.configure.rfSettings.set_envelope_power(exp_nominal_power = 1.0)
```

Sets the expected nominal power of the measured RF signal. INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:CDMA:SIGN<i>:RFPower:EPMode
- CONFIGure:CDMA:SIGN<i>:RFPower:MANual
- CONFIGure:CDMA:SIGN<i>:RFPower:EXpected

param exp_nominal_power numeric The range of the expected nominal power can be calculated as follows: Range (Expected Nominal Power) = Range (Input Power) + External Attenuation - User Margin The input power range is stated in the data sheet. Unit: dBm

set_ffset(*frequency_offset: float*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:FOFFset
driver.configure.rfSettings.set_ffset(frequency_offset = 1.0)
```

Selects a positive or negative offset frequency to be added to the center frequency (method RsCmwCdma2kMeas.Configure.RfSettings.frequency). For the combined signal path scenario, use CONFIGure:CDMA:SIGN<i>:RFSettings:FOFFset.

param frequency_offset numeric Range: -50 kHz to 50 kHz, Unit: Hz

set_frequency(*frequency: float*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:FREquency
driver.configure.rfSettings.set_frequency(frequency = 1.0)
```

Selects the center frequency of the RF analyzer. If the center frequency is valid for the current band class, the corresponding channel number is also calculated and set.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:CDMA:SIGN<i>:RFSettings:RLFrequency
- CONFIGure:CDMA:SIGN<i>:RFSettings:FREquency or
- CONFIGure:CDMA:SIGN<i>:RFSettings:CHANnel

The supported frequency range depends on the instrument model and the available options. The supported range can be smaller than stated here. Refer to the preface of your model-specific base unit manual.

param frequency numeric Range: 100 MHz to 6 GHz, Unit: Hz

set_umargin(*user_margin: float*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:RFSettings:UMARgin
driver.configure.rfSettings.set_umargin(user_margin = 1.0)
```


Sets the margin that the measurement adds to the expected nominal power to determine the reference power. The reference power minus the external input attenuation must be within the power range of the selected input connector. Refer to the data sheet.

param user_margin numeric Range: 0 dB to (55 dB + External Attenuation - Expected Nominal Power) , Unit: dB

7.2.2 MultiEval

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:TOUT
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:REPetition
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:SCONdition
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:SFACTOR
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:MOEXception
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:IQLCheck
```

class MultiEval

MultiEval commands group definition. 50 total commands, 5 Sub-groups, 6 group commands

get_iql_check() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:IQLCheck
value: bool = driver.configure.multiEval.get_iql_check()
```

Enables or disables the CDP I/Q leakage check.

return iql_leak_state: OFF | ON

get_mo_exception() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:MOEXception
value: bool = driver.configure.multiEval.get_mo_exception()
```

Specifies whether measurement results that the R&S CMW identifies as faulty or inaccurate are rejected.

return meas_on_exception: OFF | ON ON: Results are never rejected OFF: Faulty results are rejected

get_repetition() → RsCmwCdma2kMeas.enums.Repeat

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:REPetition
value: enums.Repeat = driver.configure.multiEval.get_repetition()
```

Specifies the repetition mode of the measurement. The repetition mode specifies whether the measurement is stopped after a single shot or repeated continuously. Use CONFIGure::MEAS<i>::SCOUT to determine the number of measurement intervals per single shot.

return repetition: SINGleshot | CONTInuous SINGleshot: Single-shot measurement
CONTInuous: Continuous measurement

get_scondition() → RsCmwCdma2kMeas.enums.StopCondition

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SCONdition
value: enums.StopCondition = driver.configure.multiEval.get_scondition()
```

Qualifies whether the measurement is stopped after a failed limit check or continued. SLFail means that the measurement is stopped (STOP:...MEAS<i>...) and reaches the RDY state when one of the results exceeds the limits.

return stop_condition: NONE | SLFail NONE: Continue measurement irrespective of the limit check SLFail: Stop measurement on limit failure

get_sfactor() → RsCmwCdma2kMeas.enums.SpreadingFactor

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SFActor
value: enums.SpreadingFactor = driver.configure.multiEval.get_sfactor()
```

Selects the spreading factor for the code domain power and code domain error measurements.

return spreading_factor: SF16|SF32|SF64 SF16: spreading factor 16 SF32: spreading factor 32 SF64: spreading factor 64

get_timeout() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:TOUT
value: float = driver.configure.multiEval.get_timeout()
```

Defines a timeout for the measurement. The timer is started when the measurement is initiated via a READ or INIT command. It is not started if the measurement is initiated manually ([ON | OFF] key or [RESTART | STOP] key) . When the measurement has completed the first measurement cycle (first single shot) , the statistical depth is reached and the timer is reset. If the first measurement cycle has not been completed when the timer expires, the measurement is stopped. The measurement state changes to RDY. The reliability indicator is set to 1, indicating that a measurement timeout occurred. Still running READ, FETCh or CALCulate commands are completed, returning the available results. At least for some results, there are no values at all or the statistical depth has not been reached. A timeout of 0 s corresponds to an infinite measurement timeout.

return timeout: numeric Unit: s

set_iql_check(iq_leak_state: bool) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:IQLCheck
driver.configure.multiEval.set_iql_check(iq_leak_state = False)
```

Enables or disables the CDP I/Q leakage check.

param iq_leak_state OFF | ON

set_mo_exception(meas_on_exception: bool) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:MOEXception
driver.configure.multiEval.set_mo_exception(meas_on_exception = False)
```

Specifies whether measurement results that the R&S CMW identifies as faulty or inaccurate are rejected.

param meas_on_exception OFF | ON ON: Results are never rejected OFF: Faulty results are rejected

set_repetition(*repetition*: RsCmwCdma2kMeas.enums.Repeat) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:REPetition
driver.configure.multiEval.set_repetition(repetition = enums.Repeat.CONTinuous)
```

Specifies the repetition mode of the measurement. The repetition mode specifies whether the measurement is stopped after a single shot or repeated continuously. Use CONFIGure:::MEAS<i>:::SCOunt to determine the number of measurement intervals per single shot.

param repetition SINGleshot | CONTinuous SINGleshot: Single-shot measurement
CONTinuous: Continuous measurement

set_scondition(*stop_condition*: RsCmwCdma2kMeas.enums.StopCondition) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SCONdition
driver.configure.multiEval.set_scondition(stop_condition = enums.StopCondition.
↳ NONE)
```

Qualifies whether the measurement is stopped after a failed limit check or continued. SLFail means that the measurement is stopped (STOP:...MEAS<i>...) and reaches the RDY state when one of the results exceeds the limits.

param stop_condition NONE | SLFail NONE: Continue measurement irrespective of
the limit check SLFail: Stop measurement on limit failure

set_sfactor(*spreading_factor*: RsCmwCdma2kMeas.enums.SpreadingFactor) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SFActor
driver.configure.multiEval.set_sfactor(spreading_factor = enums.SpreadingFactor.
↳ SF16)
```

Selects the spreading factor for the code domain power and code domain error measurements.

param spreading_factor SF16 | SF32 | SF64 SF16: spreading factor 16 SF32: spread-
ing factor 32 SF64: spreading factor 64

set_timeout(*timeout*: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:TOUT
driver.configure.multiEval.set_timeout(timeout = 1.0)
```

Defines a timeout for the measurement. The timer is started when the measurement is initiated via a READ or INIT command. It is not started if the measurement is initiated manually ([ON | OFF] key or [RESTART | STOP] key) . When the measurement has completed the first measurement cycle (first single shot) , the statistical depth is reached and the timer is reset. If the first measurement cycle has not been completed when the timer expires, the measurement is stopped. The measurement state changes to RDY. The reliability indicator is set to 1, indicating that a measurement timeout occurred. Still running READ, FETCH or CALCulate commands are completed, returning the available results. At least for some results, there are no values at all or the statistical depth has not been reached. A timeout of 0 s corresponds to an infinite measurement timeout.

param timeout numeric Unit: s

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.multiEval.clone()
```

Subgroups

7.2.2.1 Scount

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SCount:MODulation
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SCount:SPECTrum
```

class Scount

Scount commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

get_modulation() → int

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SCount:MODulation
value: int = driver.configure.multiEval.scount.get_modulation()
```

Specifies the statistic count of the measurement. The statistic count is equal to the number of measurement intervals per single shot.

return scount_mod: numeric Number of measurement intervals. Range: 1 to 1000

get_spectrum() → int

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SCount:SPECTrum
value: int = driver.configure.multiEval.scount.get_spectrum()
```

Specifies the statistic count of the measurement. The statistic count is equal to the number of measurement intervals per single shot.

return scount_spectrum: numeric Number of measurement intervals. Range: 1 to 1000

set_modulation(scount_mod: int) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SCount:MODulation
driver.configure.multiEval.scount.set_modulation(scount_mod = 1)
```

Specifies the statistic count of the measurement. The statistic count is equal to the number of measurement intervals per single shot.

param scount_mod numeric Number of measurement intervals. Range: 1 to 1000

set_spectrum(scount_spectrum: int) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:SCount:SPECTrum
driver.configure.multiEval.scount.set_spectrum(scount_spectrum = 1)
```

Specifies the statistic count of the measurement. The statistic count is equal to the number of measurement intervals per single shot.

param scout_spectrum numeric Number of measurement intervals. Range: 1 to 1000

7.2.2.2 ListPy

SCPI Commands

```
CONFfigure:CDMA:MEASurement<Instance>:MEValuation:LIST:COUNT
CONFfigure:CDMA:MEASurement<Instance>:MEValuation:LIST
```

class ListPy

ListPy commands group definition. 7 total commands, 2 Sub-groups, 2 group commands

get_count() → int

```
# SCPI: CONFfigure:CDMA:MEASurement<Instance>:MEValuation:LIST:COUNT
value: int = driver.configure.multiEval.listPy.get_count()
```

Defines the number of segments in the entire measurement interval.

return segments: numeric Range: 1 to 200

get_value() → bool

```
# SCPI: CONFfigure:CDMA:MEASurement<Instance>:MEValuation:LIST
value: bool = driver.configure.multiEval.listPy.get_value()
```

Enables or disables the list mode.

return enable: OFF | ON ON: Enable list mode OFF: Disable list mode

set_count(segments: int) → None

```
# SCPI: CONFfigure:CDMA:MEASurement<Instance>:MEValuation:LIST:COUNT
driver.configure.multiEval.listPy.set_count(segments = 1)
```

Defines the number of segments in the entire measurement interval.

param segments numeric Range: 1 to 200

set_value(enable: bool) → None

```
# SCPI: CONFfigure:CDMA:MEASurement<Instance>:MEValuation:LIST
driver.configure.multiEval.listPy.set_value(enable = False)
```

Enables or disables the list mode.

param enable OFF | ON ON: Enable list mode OFF: Disable list mode

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.multiEval.listPy.clone()
```

Subgroups

7.2.2.2.1 SingleCmw

SCPI Commands

```
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIST:CMWS:CMODE
```

class SingleCmw

SingleCmw commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

get_cmode() → RsCmwCdma2kMeas.enums.ParameterSetMode

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIST:CMWS:CMODE
value: enums.ParameterSetMode = driver.configure.multiEval.listPy.singleCmw.get_
↳ cmode()
```

Specifies how the input connector is selected for CDMA2000 list mode measurements with the R&S CMWS.

return cmws_connector_mode: No help available

set_cmode(cmws_connector_mode: RsCmwCdma2kMeas.enums.ParameterSetMode) → None

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIST:CMWS:CMODE
driver.configure.multiEval.listPy.singleCmw.set_cmode(cmws_connector_mode =
↳ enums.ParameterSetMode.GLOBal)
```

Specifies how the input connector is selected for CDMA2000 list mode measurements with the R&S CMWS.

param cmws_connector_mode GLOBal | LIST GLOBal: The same input connector is used for all segments. It is selected in the same way as without list mode, for example via ROUTe:CDMA:MEASi:SCENario:SALone. LIST: The input connector is configured individually for each segment. See method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.SingleCmw.Connector.set or method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Setup.set.

7.2.2.2.2 Segment<Segment>

RepCap Settings

```
# Range: Nr1 .. Nr200
rc = driver.configure.multiEval.listPy.segment.repcap_segment_get()
driver.configure.multiEval.listPy.segment.repcap_segment_set(repcap.Segment.Nr1)
```

class Segment

Segment commands group definition. 4 total commands, 4 Sub-groups, 0 group commands Repeated Capability: Segment, default value after init: Segment.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.multiEval.listPy.segment.clone()
```

Subgroups

7.2.2.2.2.1 Setup

SCPI Commands

```
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:SETup
```

class Setup

Setup commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class SetupStruct

Structure for setting input parameters. Contains optional setting parameters. Fields:

- Segment_Length: int: integer Number of measured power control groups (slots) in the segment. The sum of the segment lengths must not exceed 10000 slots; limit violation is indicated by a reliability indicator of 2 (capture buffer overflow) . Range: 1 to 1000
- Level: float: numeric Expected nominal power in the segment. The range of the expected nominal power can be calculated as follows: Range (expected nominal power) = range (input power) + external attenuation - user margin The input power range is stated in the data sheet.
- Frequency: float: numeric Center frequency of the RF analyzer. Range: 100 MHz to 6 GHz , Unit: Hz
- Retrigger_Option: enums.RettriggerOption: OFF | ON | IFPower | IFPSync Enables or disables the trigger system to start segment measurement. Note: For the first segment (no = 1) the setting of this parameter is ignored, since the general MELM measurement starts with the measurement of the first segment. That means, with the first trigger event the first segment is always measured. OFF: Disables the retrigger. The segment measurement is started by the first trigger event. ON: Enables the retrigger. The list mode measurement continues only if a new event for this segment is triggered (retrigger) . IFPower: Waits for the power ramp of the received bursts before measuring the segment. IFPSync: Before measuring the next segment, the R&S CMW waits for the power ramp of the received bursts and tries to synchronize to a slot boundary after the trigger event.

- **Rconfig:** `enums.Rconfig`: R12Q | R36H | R3Q Selects the radio configuration which determines, for example, the modulation type. Note that if `RetriggerOption=OFF` for all segments (i.e. in ‘trigger once’ mode), the radio configuration is inherited from the multi-evaluation measurement (see [CMDLINK: `CONFigure:CDMA:MEASi:RCONfig` CMDLINK]), ignoring the `RConfig` values specified for the individual segments. R12Q: RC1 or 2 (O-QPSK) R36H: RC3 to 6 (H-PSK) R3Q: RC3 (QPSK)
- **Cmws_Connector:** `enums.CmwsConnector`: Optional setting parameter. Optional parameter, as alternative to the command [CMDLINK: `CONFigure:CDMA:MEASi:MEValuation:LIST:SEGMENTno:CMWS:CONNector` CMDLINK]

get(*segment*=<*Segment.Default*: -1>) → `SetupStruct`

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGMENT<nr>:SETup
value: SetupStruct = driver.configure.multiEval.listPy.segment.setup.
↪get(segment = repcap.Segment.Default)
```

Defines the length of segment <no> and the analyzer settings. In general, this command must be sent for all segments measured.

param segment optional repeated capability selector. Default value: `Nr1` (settable in the interface ‘Segment’)

return structure: for return value, see the help for `SetupStruct` structure arguments.

set(*structure*:

RsCmwCdma2kMeas.Implementations.Configure_MultiEval_ListPy_Segment_Setup.Setup.SetupStruct,
segment=<*Segment.Default*: -1>) → `None`

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGMENT<nr>:SETup
driver.configure.multiEval.listPy.segment.setup.set(value = [PROPERTY_STRUCT_
↪NAME](), segment = repcap.Segment.Default)
```

Defines the length of segment <no> and the analyzer settings. In general, this command must be sent for all segments measured.

param structure for set value, see the help for `SetupStruct` structure arguments.

param segment optional repeated capability selector. Default value: `Nr1` (settable in the interface ‘Segment’)

7.2.2.2.2 SingleCmw

class SingleCmw

SingleCmw commands group definition. 1 total commands, 1 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.multiEval.listPy.segment.singleCmw.clone()
```

Subgroups

7.2.2.2.3 Connector

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CMWS:CONNector
```

class Connector

Connector commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

get(segment=<Segment.Default: -1>) → RsCmwCdma2kMeas.enums.CmwsConnector

```
# SCPI: CONFIGure:CDMA:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:CMWS:CONNector
value: enums.CmwsConnector = driver.configure.multiEval.listPy.segment.
↳singleCmw.connector.get(segment = repcap.Segment.Default)
```

Selects the RF input connector for segment <no> for CDMA2000 list mode measurements with the R&S CMWS. This setting is only relevant for connector mode LIST, see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.SingleCmw.cmode. All segments of a list mode measurement must use connectors of the same bench. For possible connector values, see ‘Values for RF Path Selection’.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return cmws_connector: Selects the input connector of the R&S CMWS

set(cmws_connector: RsCmwCdma2kMeas.enums.CmwsConnector, segment=<Segment.Default: -1>) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:CMWS:CONNector
driver.configure.multiEval.listPy.segment.singleCmw.connector.set(cmws_
↳connector = enums.CmwsConnector.R11, segment = repcap.Segment.Default)
```

Selects the RF input connector for segment <no> for CDMA2000 list mode measurements with the R&S CMWS. This setting is only relevant for connector mode LIST, see method

RsCmwCdma2kMeas.Configure.MultiEval.ListPy.SingleCmw.cmode. All segments of a list mode measurement must use connectors of the same bench. For possible connector values, see ‘Values for RF Path Selection’.

param cmws_connector Selects the input connector of the R&S CMWS

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

7.2.2.2.4 Modulation

SCPI Commands

CONFigure:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:MODulation

class Modulation

Modulation commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class ModulationStruct

Structure for setting input parameters. Contains optional setting parameters. Fields:

- **Statistic_Length:** int: integer The statistical length is limited by the segment length (set with [CMDLINK: CONFigure:CDMA:MEASi:MEValuation:LIST:SEGmentno:SETup CMDLINK]) and depends on the used trigger: Statistical length segment length - 2 for ‘IF Auto Sync’ trigger, Statistical length segment length - 1 for other triggers
- **Enable_Evm:** bool: OFF | ON OFF: Disable measurement. ON: Enable measurement of EVM.
- **Enable_Mag_Error:** bool: OFF | ON Disable or enable measurement of magnitude error.
- **Enable_Phase_Err:** bool: OFF | ON Disable or enable measurement of phase error.
- **Enable_Wave_Qual:** bool: OFF | ON Disable or enable measurement of waveform quality.
- **Enable_Iq_Error:** bool: OFF | ON Disable or enable measurement of I/Q origin offset and imbalance.
- **Enable_Ch_Pow:** bool: OFF | ON Disable or enable measurement of channel power.
- **Enable_Ch_Time_Off:** bool: OFF | ON Disable or enable measurement for channel time offset.
- **Enable_Ch_Phse_Off:** bool: OFF | ON Disable or enable measurement of channel phase.
- **Enable_Wbnb_Pow:** bool: OFF | ON Disable or enable measurement of wideband and narrowband power.
- **Enable_Freq_Err:** bool: OFF | ON Disable or enable measurement of carrier frequency error.
- **Enable_Melm_Tte:** bool: Optional setting parameter. OFF | ON Disable or enable measurement of transmit time error.

get(segment=<Segment.Default: -1>) → ModulationStruct

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
→:MODulation
value: ModulationStruct = driver.configure.multiEval.listPy.segment.modulation.
→get(segment = repcap.Segment.Default)
```

Defines the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and enables the calculation of the different modulation results in segment <no>; see ‘Multi-Evaluation List Mode’.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for ModulationStruct structure arguments.

set(structure:

RsCmwCdma2kMeas.Implementations.Configure_MultiEval_ListPy_Segment_Modulation.Modulation.ModulationStruct.

segment=<Segment.Default: -1>) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:MODulation
driver.configure.multiEval.listPy.segment.modulation.set(value = [PROPERTY_
↳STRUCT_NAME](), segment = repcap.Segment.Default)
```

Defines the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and enables the calculation of the different modulation results in segment <no>; see ‘Multi-Evaluation List Mode’.

param structure for set value, see the help for ModulationStruct structure arguments.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

7.2.2.2.5 Spectrum

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:SPECtrum
```

class Spectrum

Spectrum commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class SpectrumStruct

Structure for setting input parameters. Fields:

- **Statistic_Length**: int: integer The statistical length is limited by the segment length (set with [CMDLINK: CONFIGure:CDMA:MEASi:MEValuation:LIST:SEGmentno:SETup CMDLINK]) and depends on the used trigger: Statistical length segment length - 2 for ‘IF Auto Sync’ trigger, Statistical length segment length - 1 for other triggers
- **Enable_Acp_Rms**: bool: OFF | ON OFF: Disable measurement ON: Enable measurement of adjacent channel power (RMS) .
- **Enable_Obw**: bool: OFF | ON Disable or enable measurement of the occupied bandwidth.

get(segment=<Segment.Default: -1>) → SpectrumStruct

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:SPECtrum
value: SpectrumStruct = driver.configure.multiEval.listPy.segment.spectrum.
↳get(segment = repcap.Segment.Default)
```

Defines the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and enables the calculation of the different spectrum results in segment <no>; see ‘Multi-Evaluation List Mode’. Defines the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and enables the calculation of the different modulation results in segment <no>; see ‘Multi-Evaluation List Mode’.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for SpectrumStruct structure arguments.

set(*structure*:

RsCmwCdma2kMeas.Implementations.Configure_.MultiEval_.ListPy_.Segment_.Spectrum.Spectrum.SpectrumStruct,
segment=<Segment.Default: -1>) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:SPECTrum
driver.configure.multiEval.listPy.segment.spectrum.set(value = [PROPERTY_STRUCT_
↳NAME](), segment = repcap.Segment.Default)
```

Defines the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and enables the calculation of the different spectrum results in segment <no>; see ‘Multi-Evaluation List Mode’. Defines the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and enables the calculation of the different modulation results in segment <no>; see ‘Multi-Evaluation List Mode’.

param structure for set value, see the help for SpectrumStruct structure arguments.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

7.2.2.3 Acp

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:F0FFsets
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:RBW
```

class Acp

Acp commands group definition. 4 total commands, 1 Sub-groups, 2 group commands

get_foffsets() → List[float]

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:F0FFsets
value: List[float or bool] = driver.configure.multiEval.acp.get_foffsets()
```

Defines the frequency offsets to be used for ACP measurements. The offsets are defined relative to the analyzer frequency. Up to 10 offsets can be defined and enabled.

return frequency_offset: No help available

get_rbw() → List[RsCmwCdma2kMeas.enums.Rbw]

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:RBW
value: List[enums.Rbw] = driver.configure.multiEval.acp.get_rbw()
```

Defines the resolution bandwidth to be used for the upper and lower frequency offsets 0 to 9 of ACP measurements.

return rbw: F1K0 | F6K25 | F10K | F12K5 | F25K | F30K | F50K | F100k | F1M0 |
F1M23 F1K0: 1 kHz F6K25: 6.25 kHz F10K: 10 kHz F12K5: 12.5 kHz F25K: 25
kHz F30K: 30 kHz F50K: 50 kHz F100k: 100 kHz F1M0: 1 MHz F1M23: 1.23 MHz

set_foffsets(frequency_offset: List[float]) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:FOFFsets
driver.configure.multiEval.acp.set_foffsets(frequency_offset = [1.1, True, 2.2,
↪False, 3.3])
```

Defines the frequency offsets to be used for ACP measurements. The offsets are defined relative to the analyzer frequency. Up to 10 offsets can be defined and enabled.

param frequency_offset numeric | OFF | ON Range: 0 MHz to 4 MHz, Unit: MHz
Additional parameters: OFF | ON (disables the offset | enables the offset using the previous defined value)

set_rbw(rbw: List[RsCmwCdma2kMeas.enums.Rbw]) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:RBW
driver.configure.multiEval.acp.set_rbw(rbw = [Rbw.F100k, Rbw.F6K25])
```

Defines the resolution bandwidth to be used for the upper and lower frequency offsets 0 to 9 of ACP measurements.

param rbw F1K0 | F6K25 | F10K | F12K5 | F25K | F30K | F50K | F100k | F1M0 | F1M23
F1K0: 1 kHz F6K25: 6.25 kHz F10K: 10 kHz F12K5: 12.5 kHz F25K: 25 kHz F30K: 30 kHz F50K: 50 kHz F100k: 100 kHz F1M0: 1 MHz F1M23: 1.23 MHz

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.multiEval.acp.clone()
```

Subgroups

7.2.2.3.1 Extended

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:EXTended:FOFFsets
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:EXTended:RBW
```

class Extended

Extended commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

get_foffsets() → List[float]

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:EXTended:FOFFsets
value: List[float or bool] = driver.configure.multiEval.acp.extended.get_
↪foffsets()
```

Defines the frequency offsets to be used for extended ACP measurements. The offsets are defined relative to the analyzer frequency. Up to 20 offsets can be defined and enabled.

return frequency_offset: Range: 0 MHz to 4 MHz Additional parameters: OFF | ON
(disables | enables the offset)

get_rbw() → List[RsCmwCdma2kMeas.enums.Rbw]

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:EXTended:RBW
value: List[enums.Rbw] = driver.configure.multiEval.acp.extended.get_rbw()
```

Defines the resolution bandwidth to be used for the upper and lower frequency offsets 0 to 19 of extended ACP measurements.

return rbw: F1K0 | F6K25 | F10K | F12K5 | F25K | F30K | F50K | F100k | F1M0 | F1M23
F1K0: 1 kHz F6K25: 6.25 kHz F10K: 10 kHz F12K5: 12.5 kHz F25K: 25 kHz F30K: 30 kHz F50K: 50 kHz F100k: 100 kHz F1M0: 1 MHz F1M23: 1.23 MHz

set_foffsets(frequency_offset: List[float]) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:EXTended:FOFFsets
driver.configure.multiEval.acp.extended.set_foffsets(frequency_offset = [1.1, 2.2, 3.3])
↪True, 2.2, False, 3.3])
```

Defines the frequency offsets to be used for extended ACP measurements. The offsets are defined relative to the analyzer frequency. Up to 20 offsets can be defined and enabled.

param frequency_offset Range: 0 MHz to 4 MHz Additional parameters: OFF | ON
(disables | enables the offset)

set_rbw(rbw: List[RsCmwCdma2kMeas.enums.Rbw]) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:ACP:EXTended:RBW
driver.configure.multiEval.acp.extended.set_rbw(rbw = [Rbw.F100k, Rbw.F6K25])
```

Defines the resolution bandwidth to be used for the upper and lower frequency offsets 0 to 19 of extended ACP measurements.

param rbw F1K0 | F6K25 | F10K | F12K5 | F25K | F30K | F50K | F100k | F1M0 | F1M23
F1K0: 1 kHz F6K25: 6.25 kHz F10K: 10 kHz F12K5: 12.5 kHz F25K: 25 kHz F30K: 30 kHz F50K: 50 kHz F100k: 100 kHz F1M0: 1 MHz F1M23: 1.23 MHz

7.2.2.4 Result

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:ALL
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:EVMagnitude
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:MERRor
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:PERRor
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:ACP
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:OBW
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CDP
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CDE
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:POWer
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:MODQuality
```

(continues on next page)

(continued from previous page)

```

CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CP
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CPO
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CTO
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:IQ

```

class Result

Result commands group definition. 14 total commands, 0 Sub-groups, 14 group commands

class AllStruct

Structure for reading output parameters. Fields:

- Evm: bool: OFF | ON Error vector magnitude ON: Evaluate results and show the view OFF: Do not evaluate results, hide the view
- Magnitude_Error: bool: OFF | ON Magnitude error
- Phase_Error: bool: OFF | ON Phase error
- Acp: bool: OFF | ON Adjacent channel power
- Cdp: bool: OFF | ON Code domain power
- Cde: bool: OFF | ON Code domain error
- Power: bool: OFF | ON Power
- Tx_Measurements: bool: OFF | ON Modulation quality
- Channel_Power: bool: OFF | ON Channel power
- Obw: bool: OFF | ON Occupied bandwidth
- Ch_Phase_Offset: bool: OFF | ON Channel phase offset
- Ch_Time_Offset: bool: OFF | ON Channel time offset
- Iq: bool: OFF | ON IQ

get_acp() → bool

```

# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:ACP
value: bool = driver.configure.multiEval.result.get_acp()

```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...:RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_all() → AllStruct

```

# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult[:ALL]
value: AllStruct = driver.configure.multiEval.result.get_all()

```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement.

return structure: for return value, see the help for AllStruct structure arguments.

get_cde() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CDE
value: bool = driver.configure.multiEval.result.get_cde()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ..:RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_cdp() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CDP
value: bool = driver.configure.multiEval.result.get_cdp()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ..:RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_cp() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CP
value: bool = driver.configure.multiEval.result.get_cp()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ..:RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_cpo() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CPO
value: bool = driver.configure.multiEval.result.get_cpo()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ..:RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_cto() → bool


```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CTO
value: bool = driver.configure.multiEval.result.get_cto()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_ev_magnitude() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:EVMagnitude
value: bool = driver.configure.multiEval.result.get_ev_magnitude()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_iq() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:IQ
value: bool = driver.configure.multiEval.result.get_iq()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_merror() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:MERRor
value: bool = driver.configure.multiEval.result.get_merror()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_mod_quality() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:MODQuality
value: bool = driver.configure.multiEval.result.get_mod_quality()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after `...RESult` denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_obw() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:OBW
value: bool = driver.configure.multiEval.result.get_obw()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after `...RESult` denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_perror() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:PERRor
value: bool = driver.configure.multiEval.result.get_perror()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after `...RESult` denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

get_power() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:POWER
value: bool = driver.configure.multiEval.result.get_power()
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after `...RESult` denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

return enable: OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_acp(enable: bool) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:ACP
driver.configure.multiEval.result.set_acp(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after `...RESult` denotes the view type: Error vector magnitude, magnitude error,

phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_all(value: *RsCmwCdma2kMeas.Implementations.Configure_MultiEval_Result.Result.AllStruct*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult[:ALL]
driver.configure.multiEval.result.set_all(value = AllStruct())
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement.

param value see the help for AllStruct structure arguments.

set_cde(enable: *bool*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CDE
driver.configure.multiEval.result.set_cde(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ..:RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_cdp(enable: *bool*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CDP
driver.configure.multiEval.result.set_cdp(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ..:RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_cp(enable: *bool*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CP
driver.configure.multiEval.result.set_cp(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ..:RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_cpo(*enable: bool*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CPO
driver.configure.multiEval.result.set_cpo(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_cto(*enable: bool*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:CTO
driver.configure.multiEval.result.set_cto(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_ev_magnitude(*enable: bool*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:EVMagnitude
driver.configure.multiEval.result.set_ev_magnitude(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_iq(*enable: bool*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:IQ
driver.configure.multiEval.result.set_iq(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_merror(*enable: bool*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:MERRor
driver.configure.multiEval.result.set_merror(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_mod_quality(enable: bool) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:MODQuality
driver.configure.multiEval.result.set_mod_quality(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_obw(enable: bool) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:OBW
driver.configure.multiEval.result.set_obw(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_perror(enable: bool) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:PERRor
driver.configure.multiEval.result.set_perror(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...RESult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

set_power(enable: bool) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:RESult:POWER
driver.configure.multiEval.result.set_power(enable = False)
```

Enables or disables the evaluation of results and shows or hides the views in the multi-evaluation measurement. The mnemonic after ...REsult denotes the view type: Error vector magnitude, magnitude error, phase error, adjacent channel power, code domain power, code domain error, channel power, channel phase offset, channel time offset, IQ, power and modulation quality.

param enable OFF | ON ON: Evaluate results and show view OFF: Do not evaluate results, hide view

7.2.2.5 Limit

SCPI Commands

```

CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:POWer
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:EVMagnitude
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:MERRor
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:PERRor
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:IQOffset
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:IQIMbalance
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:CFERror
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:TTERror
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:WFQuality
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:OBW
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:CDP
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:CDE
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:CP
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:CPO
CONFigure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:CTO

```

class Limit

Limit commands group definition. 17 total commands, 1 Sub-groups, 15 group commands

class EvMagnitudeStruct

Structure for reading output parameters. Fields:

- Evm_Rms: float or bool: Range: 0 % to 100 %, Unit: % Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)
- Evm_Peak: float or bool: Range: 0 % to 100 %, Unit: % Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

class MerrorStruct

Structure for reading output parameters. Fields:

- Merr_Rms: float or bool: Range: 0 % to 100 %, Unit: % Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)
- Merr_Peak: float or bool: Range: 0 % to 100 %, Unit: % Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

class PerrorStruct

Structure for reading output parameters. Fields:

- Perr_Rms: float or bool: Range: 0 deg to 180 deg, Unit: deg Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)
- Perr_Peak: float or bool: Range: 0 deg to 180 deg, Unit: deg Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

class PowerStruct

Structure for reading output parameters. Fields:

- **Min_Power:** float or bool: Range: -128 dBm to 50 dBm, Unit: dBm Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)
- **Max_Power:** float or bool: Range: -128 dBm to 50 dBm, Unit: dBm Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_cde() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CDE
value: float or bool = driver.configure.multiEval.limit.get_cde()
```

Defines an upper limit for the code domain error.

return cdep: Range: -70 dB to 0 dB, Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_cdp() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CDP
value: float or bool = driver.configure.multiEval.limit.get_cdp()
```

Defines an upper limit for the code domain power of inactive channels.

return limit_cdp: Range: -70 dB to 0 dB, Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_cf_error() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CFError
value: float or bool = driver.configure.multiEval.limit.get_cf_error()
```

Defines an upper limit for the carrier frequency error.

return cf_error: Range: 0 Hz to 1000 Hz, Unit: Hz Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_cp() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CP
value: float or bool = driver.configure.multiEval.limit.get_cp()
```

Defines an upper limit for the channel power.

return channel_power: Limit of the channel power. Range: -60 dB to 0 dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_cpo() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CPO
value: float or bool = driver.configure.multiEval.limit.get_cpo()
```

Defines an upper limit for the absolute channel phase offset.

return ch_phase_offset: Limit for absolute phase offset. Range: 0 mRad to 200 mRad, Unit: mRad Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_cto() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CTO
value: float or bool = driver.configure.multiEval.limit.get_cto()
```

Defines the upper limit for the absolute time offset.

return ch_time_offset: Limit for absolute time offset in nanoseconds. Range: 0 ns to 40 ns, Unit: ns

get_ev_magnitude() → EvMagnitudeStruct

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:EVMagnitude
value: EvMagnitudeStruct = driver.configure.multiEval.limit.get_ev_magnitude()
```

Defines upper limits for the RMS and peak values of the error vector magnitude (EVM) .

return structure: for return value, see the help for EvMagnitudeStruct structure arguments.

get_iq_imbalance() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:IQImbalance
value: float or bool = driver.configure.multiEval.limit.get_iq_imbalance()
```

Defines an upper limit for the I/Q imbalance.

return iq_imbalance: Range: -120 dB to -20 dB, Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_iq_offset() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:IQOffset
value: float or bool = driver.configure.multiEval.limit.get_iq_offset()
```

Defines an upper limit for the I/Q origin offset (carrier feedthrough) .

return iq_offset: Range: -120 dB to -20 dB, Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_merror() → MerrorStruct


```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:MErRor
value: MerrorStruct = driver.configure.multiEval.limit.get_merror()
```

Defines upper limits for the RMS and peak values of the magnitude error.

return structure: for return value, see the help for MerrorStruct structure arguments.

get_obw() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:OBW
value: float or bool = driver.configure.multiEval.limit.get_obw()
```

Defines an upper limit for the occupied bandwidth.

return limit_obw: Range: 0 MHz to 8 MHz, Unit: MHz Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_perror() → PerrorStruct

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:PERRor
value: PerrorStruct = driver.configure.multiEval.limit.get_perror()
```

Defines a symmetric limit for the RMS and peak values of the phase error. The limit check fails if the absolute value of the measured phase error exceeds the specified value.

return structure: for return value, see the help for PerrorStruct structure arguments.

get_power() → PowerStruct

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:POWer
value: PowerStruct = driver.configure.multiEval.limit.get_power()
```

Defines limits for the mobile station (MS) power.

return structure: for return value, see the help for PowerStruct structure arguments.

get_tt_error() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:TTErRor
value: float or bool = driver.configure.multiEval.limit.get_tt_error()
```

Defines an upper limit for the transport time error.

return tt_error: Range: 0 μs to 10 μs, Unit: μs Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_wf_quality() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:WFQuality
value: float or bool = driver.configure.multiEval.limit.get_wf_quality()
```

Defines a lower limit for the waveform quality. For an ideal transmitter, the waveform quality equals 1.

return wf_qual: Range: 0 to 1 Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_cde(cdep: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CDE
driver.configure.multiEval.limit.set_cde(cdep = 1.0)
```

Defines an upper limit for the code domain error.

param cdep Range: -70 dB to 0 dB, Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_cdp(limit_cdp: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CDP
driver.configure.multiEval.limit.set_cdp(limit_cdp = 1.0)
```

Defines an upper limit for the code domain power of inactive channels.

param limit_cdp Range: -70 dB to 0 dB, Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_cf_error(cf_error: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CFError
driver.configure.multiEval.limit.set_cf_error(cf_error = 1.0)
```

Defines an upper limit for the carrier frequency error.

param cf_error Range: 0 Hz to 1000 Hz, Unit: Hz Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_cp(channel_power: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CP
driver.configure.multiEval.limit.set_cp(channel_power = 1.0)
```

Defines an upper limit for the channel power.

param channel_power Limit of the channel power. Range: -60 dB to 0 dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_cpo(ch_phase_offset: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CPO
driver.configure.multiEval.limit.set_cpo(ch_phase_offset = 1.0)
```

Defines an upper limit for the absolute channel phase offset.

param ch_phase_offset Limit for absolute phase offset. Range: 0 mRad to 200 mRad, Unit: mRad Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_cto(ch_time_offset: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:CTO
driver.configure.multiEval.limit.set_cto(ch_time_offset = 1.0)
```

Defines the upper limit for the absolute time offset.

param ch_time_offset Limit for absolute time offset in nanoseconds. Range: 0 ns to 40 ns, Unit: ns

set_ev_magnitude(value: RsCmwCdma2kMeas.Implementations.Configure_MultiEval_Limit.Limit.EvMagnitudeStruct) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:EVMagnitude
driver.configure.multiEval.limit.set_ev_magnitude(value = EvMagnitudeStruct())
```

Defines upper limits for the RMS and peak values of the error vector magnitude (EVM) .

param value see the help for EvMagnitudeStruct structure arguments.

set_iq_imbalance(iq_imbalance: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:IQIMbalance
driver.configure.multiEval.limit.set_iq_imbalance(iq_imbalance = 1.0)
```

Defines an upper limit for the I/Q imbalance.

param iq_imbalance Range: -120 dB to -20 dB, Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_iq_offset(iq_offset: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:IQOffset
driver.configure.multiEval.limit.set_iq_offset(iq_offset = 1.0)
```

Defines an upper limit for the I/Q origin offset (carrier feedthrough) .

param iq_offset Range: -120 dB to -20 dB, Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_merror(value: RsCmwCdma2kMeas.Implementations.Configure_MultiEval_Limit.Limit.MerrorStruct) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:MERROR
driver.configure.multiEval.limit.set_merror(value = MerrorStruct())
```

Defines upper limits for the RMS and peak values of the magnitude error.

param value see the help for MerrorStruct structure arguments.

set_obw(*limit_obw: float*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:OBW
driver.configure.multiEval.limit.set_obw(limit_obw = 1.0)
```

Defines an upper limit for the occupied bandwidth.

param limit_obw Range: 0 MHz to 8 MHz, Unit: MHz Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_perror(*value: RsCmwCdma2kMeas.Implementations.Configure_MultiEval_Limit.Limit.PerrorStruct*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:PERRor
driver.configure.multiEval.limit.set_perror(value = PerrorStruct())
```

Defines a symmetric limit for the RMS and peak values of the phase error. The limit check fails if the absolute value of the measured phase error exceeds the specified value.

param value see the help for PerrorStruct structure arguments.

set_power(*value: RsCmwCdma2kMeas.Implementations.Configure_MultiEval_Limit.Limit.PowerStruct*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:POWer
driver.configure.multiEval.limit.set_power(value = PowerStruct())
```

Defines limits for the mobile station (MS) power.

param value see the help for PowerStruct structure arguments.

set_tt_error(*tt_error: float*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:TTErroR
driver.configure.multiEval.limit.set_tt_error(tt_error = 1.0)
```

Defines an upper limit for the transport time error.

param tt_error Range: 0 μs to 10 μs, Unit: μs Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_wf_quality(*wf_qual: float*) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:WFQuality
driver.configure.multiEval.limit.set_wf_quality(wf_qual = 1.0)
```

Defines a lower limit for the waveform quality. For an ideal transmitter, the waveform quality equals 1.

param wf_qual Range: 0 to 1 Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.multiEval.limit.clone()
```

Subgroups

7.2.2.5.1 Acp

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:ACP:RELative
CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:ACP:ABSolute
```

class Acp

Acp commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

get_absolute() → List[float]

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:ACP:ABSolute
value: List[float or bool] = driver.configure.multiEval.limit.acp.get_absolute()
```

Defines limits for the ACP in dBm at the individual offset frequencies (set via method RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets).

return limit_acp: numeric Range: -80 dBm to 10 dBm , Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_relative() → List[float]

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:ACP[:RELative]
value: List[float or bool] = driver.configure.multiEval.limit.acp.get_relative()
```

Defines limits for the ACP in dBc at the individual offset frequencies (set via method RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets).

return limit_acp: No help available

set_absolute(limit_acp: List[float]) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEValuation:LIMit:ACP:ABSolute
driver.configure.multiEval.limit.acp.set_absolute(limit_acp = [1.1, True, 2.2,
↪False, 3.3])
```

Defines limits for the ACP in dBm at the individual offset frequencies (set via method RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets).

param limit_acp numeric Range: -80 dBm to 10 dBm , Unit: dB Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_relative(limit_acp: List[float]) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:LIMit:ACP[:RELative]
driver.configure.multiEval.limit.acp.set_relative(limit_acp = [1.1, True, 2.2,
↪False, 3.3])
```

Defines limits for the ACP in dBc at the individual offset frequencies (set via method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Offsets).

param limit_acp numeric Range: -80 dBc to 10 dBc , Unit: dBc Additional parameters:
OFF | ON (disables the limit check | enables the limit check using the previous/default
limit values)

7.2.3 Oltr

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:OLTR:TOUT
CONFIGure:CDMA:MEASurement<Instance>:OLTR:REPetition
CONFIGure:CDMA:MEASurement<Instance>:OLTR:SEquence
CONFIGure:CDMA:MEASurement<Instance>:OLTR:MOEXception
```

class Oltr

Oltr commands group definition. 11 total commands, 4 Sub-groups, 4 group commands

get_mo_exception() → bool

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:MOEXception
value: bool = driver.configure.oltr.get_mo_exception()
```

Specifies whether measurement results that the R&S CMW identifies as faulty or inaccurate are rejected.

return meas_on_exception: OFF | ON ON: Results are never rejected OFF: Faulty re-
sults are rejected

get_repetition() → RsCmwCdma2kMeas.enums.Repeat

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:REPetition
value: enums.Repeat = driver.configure.oltr.get_repetition()
```

Specifies the repetition mode of the measurement. The repetition mode specifies whether the measurement is stopped after a single shot or repeated continuously. Use CONFIGure:...:MEAS<i>:...:SCOut to determine the number of measurement intervals per single shot.

return repetition: SINGleshot | CONTInuous SINGleshot: Single-shot measurement
CONTInuous: Continuous measurement

get_sequence() → int

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:SEquence
value: int = driver.configure.oltr.get_sequence()
```

Sets/gets the number of measurement sequences within a single OLTR measurement. Each sequence consists of a power UP or power DOWN step, followed by a power step in the opposite direction (see method RsCmwCdma2kMeas.Configure.Oltr.Pstep. direction).

return no_of_meas_seq: numeric Range: 1 to 5

get_timeout() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:TOUT
value: float = driver.configure.oltr.get_timeout()
```

Defines a timeout for the measurement. The timer is started when the measurement is initiated via a READ or INIT command. It is not started if the measurement is initiated manually ([ON | OFF] key or [RESTART | STOP] key) . When the measurement has completed the first measurement cycle (first single shot) , the statistical depth is reached and the timer is reset. If the first measurement cycle has not been completed when the timer expires, the measurement is stopped. The measurement state changes to RDY. The reliability indicator is set to 1, indicating that a measurement timeout occurred. Still running READ, FETCh or CALCulate commands are completed, returning the available results. At least for some results, there are no values at all or the statistical depth has not been reached. A timeout of 0 s corresponds to an infinite measurement timeout.

return timeout: numeric Unit: s

set_mo_exception(meas_on_exception: bool) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:MOEXception
driver.configure.oltr.set_mo_exception(meas_on_exception = False)
```

Specifies whether measurement results that the R&S CMW identifies as faulty or inaccurate are rejected.

param meas_on_exception OFF | ON ON: Results are never rejected OFF: Faulty results are rejected

set_sequence(no_of_meas_seq: int) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:SEquence
driver.configure.oltr.set_sequence(no_of_meas_seq = 1)
```

Sets/gets the number of measurement sequences within a single OLTR measurement. Each sequence consists of a power UP or power DOWN step, followed by a power step in the opposite direction (see method RsCmwCdma2kMeas.Configure.Oltr.Pstep. direction).

param no_of_meas_seq numeric Range: 1 to 5

set_timeout(timeout: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:TOUT
driver.configure.oltr.set_timeout(timeout = 1.0)
```

Defines a timeout for the measurement. The timer is started when the measurement is initiated via a READ or INIT command. It is not started if the measurement is initiated manually ([ON | OFF] key or [RESTART | STOP] key) . When the measurement has completed the first measurement cycle (first single shot) , the statistical depth is reached and the timer is reset. If the first measurement cycle has not been completed when the timer expires, the measurement is stopped. The measurement state changes to RDY. The reliability indicator is set to 1, indicating that a measurement timeout occurred. Still running READ, FETCh or

CALCulate commands are completed, returning the available results. At least for some results, there are no values at all or the statistical depth has not been reached. A timeout of 0 s corresponds to an infinite measurement timeout.

param timeout numeric Unit: s

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.configure.oltr.clone()
```

Subgroups

7.2.3.1 Pstep

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:OLTR:PSTep:DIRection
CONFIGure:CDMA:MEASurement<Instance>:OLTR:PSTep
```

class Pstep

Pstep commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

get_direction() → RsCmwCdma2kMeas.enums.UpDownDirection

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:PSTep:DIRection
value: enums.UpDownDirection = driver.configure.oltr.pstep.get_direction()
```

Defines the direction of the first power step within an OLTR measurement. For each subsequent power step, the direction is toggled.

return pstep_direction: DOWN | UP

get_value() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:PSTep
value: float = driver.configure.oltr.pstep.get_value()
```

Defines the size of the power steps, i.e. the increases and decreases in the total BSS power during the OLTR measurement.

return power_step: numeric The power step is relative to the measured reference power.
Range: 0 dB to 40 dB , Unit: dB

set_direction(pstep_direction: RsCmwCdma2kMeas.enums.UpDownDirection) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:PSTep:DIRection
driver.configure.oltr.pstep.set_direction(pstep_direction = enums.
↪UpDownDirection.DOWN)
```

Defines the direction of the first power step within an OLTR measurement. For each subsequent power step, the direction is toggled.

param pstep_direction DOWN | UP

set_value(power_step: float) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:PSTep
driver.configure.oltr.pstep.set_value(power_step = 1.0)
```

Defines the size of the power steps, i.e. the increases and decreases in the total BSS power during the OLTR measurement.

param power_step numeric The power step is relative to the measured reference power.
Range: 0 dB to 40 dB , Unit: dB

7.2.3.2 RpInterval

SCPI Commands

```
CONFIGure:CDMA:MEASurement<Instance>:OLTR:RPInterval:TIME
CONFIGure:CDMA:MEASurement<Instance>:OLTR:RPInterval
```

class RpInterval

RpInterval commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

get_time() → float

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:RPInterval:TIME
value: float = driver.configure.oltr.rpInterval.get_time()
```

Gets the duration of the reference power interval, i.e. the interval that is used to calculate the MS reference power for the subsequent power step.

return ref_pow_interval: float Range: 5 ms to 40 ms , Unit: ms

get_value() → int

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:RPInterval
value: int = driver.configure.oltr.rpInterval.get_value()
```

Gets the duration of the reference power interval, i.e. the interval that is used to calculate the MS reference power for the subsequent power step.

return ref_pow_interval: integer The time as the number of power control groups: from
4 (= 5 ms) to 32 (=40 ms) Range: 4 to 32

set_value(ref_pow_interval: int) → None

```
# SCPI: CONFIGure:CDMA:MEASurement<Instance>:OLTR:RPInterval
driver.configure.oltr.rpInterval.set_value(ref_pow_interval = 1)
```

Gets the duration of the reference power interval, i.e. the interval that is used to calculate the MS reference power for the subsequent power step.

param ref_pow_interval integer The time as the number of power control groups: from
4 (= 5 ms) to 32 (=40 ms) Range: 4 to 32

7.2.3.3 Ginterval

SCPI Commands

```
CONFigure:CDMA:MEASurement<Instance>:OLTR:GINTerval:TIME
CONFigure:CDMA:MEASurement<Instance>:OLTR:GINTerval
```

class Ginterval

Ginterval commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

get_time() → float

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:OLTR:GINTerval:TIME
value: float = driver.configure.oltr.ginterval.get_time()
```

Gets the duration of the guard intervals, i.e. the intervals succeeding the OLTR evaluation intervals and preceding the reference power intervals.

return guard_interval: Range: 0 ms to 100 ms

get_value() → int

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:OLTR:GINTerval
value: int = driver.configure.oltr.ginterval.get_value()
```

Defines the duration of the guard intervals, i.e. the intervals succeeding the OLTR evaluation intervals and preceding the reference power intervals.

return guard_interval: integer The duration of the guard interval as number of power control groups (1.25 ms) . Range: 0 to 80

set_value(guard_interval: int) → None

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:OLTR:GINTerval
driver.configure.oltr.ginterval.set_value(guard_interval = 1)
```

Defines the duration of the guard intervals, i.e. the intervals succeeding the OLTR evaluation intervals and preceding the reference power intervals.

param guard_interval integer The duration of the guard interval as number of power control groups (1.25 ms) . Range: 0 to 80

7.2.3.4 Limit

SCPI Commands

```
CONFigure:CDMA:MEASurement<Instance>:OLTR:LIMit:ILOWer
```

class Limit

Limit commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

get_ilower() → int

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:OLTR:LIMit:ILOWer
value: int = driver.configure.oltr.limit.get_ilower()
```

Sets initial lower limits for open loop power control step response (3GPP2 C.S0011) .

return initial_lower: numeric Range: -2 dB to -1 dB, Unit: dB

set_ilower(initial_lower: int) → None

```
# SCPI: CONFigure:CDMA:MEASurement<Instance>:OLTR:LIMit:ILOWer
driver.configure.oltr.limit.set_ilower(initial_lower = 1)
```

Sets initial lower limits for open loop power control step response (3GPP2 C.S0011) .

param initial_lower numeric Range: -2 dB to -1 dB, Unit: dB

7.3 Trigger

class Trigger

Trigger commands group definition. 9 total commands, 1 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.trigger.clone()
```

Subgroups

7.3.1 MultiEval

SCPI Commands

```
TRIGger:CDMA:MEASurement<Instance>:MEvaluation:SOURce
TRIGger:CDMA:MEASurement<Instance>:MEvaluation:TOUT
TRIGger:CDMA:MEASurement<Instance>:MEvaluation:SLOPe
TRIGger:CDMA:MEASurement<Instance>:MEvaluation:THReshold
TRIGger:CDMA:MEASurement<Instance>:MEvaluation:MGAP
TRIGger:CDMA:MEASurement<Instance>:MEvaluation:DElay
TRIGger:CDMA:MEASurement<Instance>:MEvaluation:EOffset
```

class MultiEval

MultiEval commands group definition. 9 total commands, 2 Sub-groups, 7 group commands

get_delay() → float

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:DElay
value: float = driver.trigger.multiEval.get_delay()
```

Defines a time delaying the start of the measurement relative to the trigger event. This setting has no influence on 'Free Run' measurements.

return delay: numeric Range: -1.25E-3 s to 0.08 s, Unit: s

get_eoffset() → int

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:EOffset
value: int = driver.trigger.multiEval.get_eoffset()
```

Defines a delay time for the measurement relative to the 'IF Power' or external trigger events (see method RsCmwCdma2kMeas.Trigger.MultiEval.source) . The range is entered as an integer number of power control groups (PCG) . Each PCG has a duration of 1.25 ms.

return eval_offset: integer Range: 0 to 64, Unit: power control group

get_mgap() → float

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:MGAP
value: float = driver.trigger.multiEval.get_mgap()
```

Sets a minimum time during which the IF signal must be below the trigger threshold before the trigger is armed so that an IF power trigger event can be generated.

return min_trigger_gap: numeric Range: 0 s to 0.01 s, Unit: s

get_slope() → RsCmwCdma2kMeas.enums.Slope

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:SLOPe
value: enums.Slope = driver.trigger.multiEval.get_slope()
```

Qualifies whether the trigger event is generated at the rising or at the falling edge of the trigger pulse (valid for external and power trigger sources) .

return slope: REDGe | FEDGe REDGe: Rising edge FEDGe: Falling edge

get_source() → str

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:SOURce
value: str = driver.trigger.multiEval.get_source()
```

Selects the source of the trigger events. Some values are always available. They are listed below. Depending on the installed options, additional values are available. You can query a list of all supported values via TRIGger:... :CATalog:SOURce?.

return trigger_name: string 'Free Run': Free run (untriggered) 'IF Power': Power trigger (received RF power) 'IF Auto Sync': Power trigger auto synchronized

get_threshold() → float

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:THReshold
value: float or bool = driver.trigger.multiEval.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return threshold: Range: -50 dB to 0 dB, Unit: dB

set_timeout() → float

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:TOUT
value: float or bool = driver.trigger.multiEval.get_timeout()
```

Selects the maximum time that the measurement waits for a trigger event before it stops in remote control mode or indicates a trigger timeout in manual operation mode. This setting has no influence on 'Free Run' measurements.

return time: Range: 0 s to 83.88607E+3 s, Unit: s Additional values: OFF | ON (disables timeout | enables timeout using the previous/default values)

set_delay(delay: float) → None

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:DELAY
driver.trigger.multiEval.set_delay(delay = 1.0)
```

Defines a time delaying the start of the measurement relative to the trigger event. This setting has no influence on 'Free Run' measurements.

param delay numeric Range: -1.25E-3 s to 0.08 s, Unit: s

set_eoffset(eval_offset: int) → None

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:EOffset
driver.trigger.multiEval.set_eoffset(eval_offset = 1)
```

Defines a delay time for the measurement relative to the 'IF Power' or external trigger events (see method RsCmwCdma2kMeas.Trigger.MultiEval.source) . The range is entered as an integer number of power control groups (PCG) . Each PCG has a duration of 1.25 ms.

param eval_offset integer Range: 0 to 64, Unit: power control group

set_mgap(min_trigger_gap: float) → None

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:MGAP
driver.trigger.multiEval.set_mgap(min_trigger_gap = 1.0)
```

Sets a minimum time during which the IF signal must be below the trigger threshold before the trigger is armed so that an IF power trigger event can be generated.

param min_trigger_gap numeric Range: 0 s to 0.01 s, Unit: s

set_slope(slope: RsCmwCdma2kMeas.enums.Slope) → None

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:SLOPe
driver.trigger.multiEval.set_slope(slope = enums.Slope.FEDGE)
```

Qualifies whether the trigger event is generated at the rising or at the falling edge of the trigger pulse (valid for external and power trigger sources) .

param slope REDGe | FEDGe REDGe: Rising edge FEDGe: Falling edge

set_source(*trigger_name: str*) → None

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:SOURce
driver.trigger.multiEval.set_source(trigger_name = '1')
```

Selects the source of the trigger events. Some values are always available. They are listed below. Depending on the installed options, additional values are available. You can query a list of all supported values via TRIGger:... :CATalog:SOURce?.

param trigger_name string ‘Free Run’: Free run (untriggered) ‘IF Power’: Power trigger (received RF power) ‘IF Auto Sync’: Power trigger auto synchronized

set_threshold(*threshold: float*) → None

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:THReshold
driver.trigger.multiEval.set_threshold(threshold = 1.0)
```

Defines the trigger threshold for power trigger sources.

param threshold Range: -50 dB to 0 dB, Unit: dB

set_timeout(*time: float*) → None

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEvaluation:TOUT
driver.trigger.multiEval.set_timeout(time = 1.0)
```

Selects the maximum time that the measurement waits for a trigger event before it stops in remote control mode or indicates a trigger timeout in manual operation mode. This setting has no influence on ‘Free Run’ measurements.

param time Range: 0 s to 83.88607E+3 s, Unit: s Additional values: OFF | ON (disables timeout | enables timeout using the previous/default values)

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.trigger.multiEval.clone()
```

Subgroups

7.3.1.1 Catalog

SCPI Commands

```
TRIGger:CDMA:MEASurement<Instance>:MEvaluation:CATalog:SOURce
```

class Catalog

Catalog commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

get_source() → List[str]

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEValuation:CATalog:SOURce
value: List[str] = driver.trigger.multiEval.catalog.get_source()
```

Lists all trigger source values that can be set using method RsCmwCdma2kMeas.Trigger.MultiEval.source.

return trigger_list: string Comma-separated list of all supported values. Each value is represented as a string.

7.3.1.2 ListPy

SCPI Commands

```
TRIGger:CDMA:MEASurement<Instance>:MEValuation:LIST:MODE
```

class ListPy

ListPy commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

get_mode() → RsCmwCdma2kMeas.enums.RetriggerMode

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEValuation:LIST:MODE
value: enums.RetriggerMode = driver.trigger.multiEval.listPy.get_mode()
```

Specifies whether a trigger event initiates a measurement of the entire measurement interval (comprising the number of segments defined via method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count) or the retrigger information from the segments is used.

return retrigger_mode: ONCE | SEGment ONCE: Trigger only once. Every segment is measured irrespective of the setting of the parameter RetriggerOption from the segment (method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Setup.set). The trigger is rearmed only after the measurement is stopped and restarted. SEGM: The measurement starts after the first trigger event and continues as long as no segment is reached that requires a retrigger (method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Setup.set). This mode is recommended for statistic counts where retriggering can compensate a possible time drift of the MS.

set_mode(retrigger_mode: RsCmwCdma2kMeas.enums.RetriggerMode) → None

```
# SCPI: TRIGger:CDMA:MEASurement<Instance>:MEValuation:LIST:MODE
driver.trigger.multiEval.listPy.set_mode(retrigger_mode = enums.RetriggerMode.
    ONCE)
```

Specifies whether a trigger event initiates a measurement of the entire measurement interval (comprising the number of segments defined via method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count) or the retrigger information from the segments is used.

param retrigger_mode ONCE | SEGment ONCE: Trigger only once. Every segment is measured irrespective of the setting of the parameter RetriggerOption from the segment (method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Setup.set). The trigger is rearmed only after the measurement is stopped and restarted. SEGM: The measurement starts after the first trigger event and continues as long as no segment is reached that requires a retrigger (method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Setup.set). This mode is

recommended for statistic counts where retriggering can compensate a possible time drift of the MS.

7.4 MultiEval

SCPI Commands

```
INITiate:CDMA:MEASurement<Instance>:MEvaluation
ABORT:CDMA:MEASurement<Instance>:MEvaluation
STOP:CDMA:MEASurement<Instance>:MEvaluation
```

class MultiEval

MultiEval commands group definition. 674 total commands, 6 Sub-groups, 3 group commands

abort() → None

```
# SCPI: ABORT:CDMA:MEASurement<Instance>:MEvaluation
driver.multiEval.abort()

INTRO_CMD_HELP: Starts, stops, or aborts the measurement:

- INITiate... starts or restarts the measurement. The measurement enters
↳ the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY
↳ ' state. Measurement results are kept. The resources remain allocated to the
↳ measurement.
- ABORT... halts the measurement immediately. The measurement enters the
↳ 'OFF' state. All measurement values are set to NAV. Allocated resources are
↳ released.
```

Use FETCh...STATe? to query the current measurement state.

abort_with_opc() → None

```
# SCPI: ABORT:CDMA:MEASurement<Instance>:MEvaluation
driver.multiEval.abort_with_opc()

INTRO_CMD_HELP: Starts, stops, or aborts the measurement:

- INITiate... starts or restarts the measurement. The measurement enters
↳ the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY
↳ ' state. Measurement results are kept. The resources remain allocated to the
↳ measurement.
- ABORT... halts the measurement immediately. The measurement enters the
↳ 'OFF' state. All measurement values are set to NAV. Allocated resources are
↳ released.
```

Use FETCh...STATe? to query the current measurement state.

Same as abort, but waits for the operation to complete before continuing further. Use the RsCmwCdma2kMeas.utilities.opc_timeout_set() to set the timeout value.

initiate() → None

```
# SCPI: INITiate:CDMA:MEASurement<Instance>:MEvaluation
```

```
driver.multiEval.initiate()
```

INTRO_CMD_HELP: Starts, stops, **or** aborts the measurement:

- INITiate... starts **or** restarts the measurement. The measurement enters the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY' state. Measurement results are kept. The resources remain allocated to the measurement.
- ABORT... halts the measurement immediately. The measurement enters the 'OFF' state. All measurement values are **set** to NAV. Allocated resources are released.

Use FETCh...STATe? to query the current measurement state.

initiate_with_opc() → None

```
# SCPI: INITiate:CDMA:MEASurement<Instance>:MEvaluation
```

```
driver.multiEval.initiate_with_opc()
```

INTRO_CMD_HELP: Starts, stops, **or** aborts the measurement:

- INITiate... starts **or** restarts the measurement. The measurement enters the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY' state. Measurement results are kept. The resources remain allocated to the measurement.
- ABORT... halts the measurement immediately. The measurement enters the 'OFF' state. All measurement values are **set** to NAV. Allocated resources are released.

Use FETCh...STATe? to query the current measurement state.

Same as initiate, but waits for the operation to complete before continuing further. Use the RsCmwCdma2kMeas.utilities.opc_timeout_set() to set the timeout value.

stop() → None

```
# SCPI: STOP:CDMA:MEASurement<Instance>:MEvaluation
```

```
driver.multiEval.stop()
```

INTRO_CMD_HELP: Starts, stops, **or** aborts the measurement:

- INITiate... starts **or** restarts the measurement. The measurement enters the 'RUN' state.

(continues on next page)

(continued from previous page)

```

- STOP... halts the measurement immediately. The measurement enters the 'RDY
↪' state. Measurement results are kept. The resources remain allocated to the
↪measurement.
- ABORT... halts the measurement immediately. The measurement enters the
↪'OFF' state. All measurement values are set to NAV. Allocated resources are
↪released.

```

Use FETCh...STATe? to query the current measurement state.

stop_with_opc() → None

```

# SCPI: STOP:CDMA:MEASurement<Instance>:MEvaluation
driver.multiEval.stop_with_opc()

```

INTRO_CMD_HELP: Starts, stops, or aborts the measurement:

```

- INITiate... starts or restarts the measurement. The measurement enters
↪the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY
↪' state. Measurement results are kept. The resources remain allocated to the
↪measurement.
- ABORT... halts the measurement immediately. The measurement enters the
↪'OFF' state. All measurement values are set to NAV. Allocated resources are
↪released.

```

Use FETCh...STATe? to query the current measurement state.

Same as stop, but waits for the operation to complete before continuing further. Use the RsCmwCdma2kMeas.utilities.opc_timeout_set() to set the timeout value.

Cloning the Group

```

# Create a clone of the original group, that exists independently
group2 = driver.multiEval.clone()

```

Subgroups

7.4.1 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:STATe
```

class State

State commands group definition. 2 total commands, 1 Sub-groups, 1 group commands

fetch() → RsCmwCdma2kMeas.enums.ResourceState

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:STATe
value: enums.ResourceState = driver.multiEval.state.fetch()
```

Queries the main measurement state. Use FETCh:...:STATe:ALL? to query the measurement state including the substates. Use INITiate..., STOP..., ABORT... to change the measurement state.

return meas_state: OFF | RUN | RDY OFF: measurement switched off, no resources allocated, no results available (when entered after ABORT...) RUN: measurement running (after INITiate..., READ...) , synchronization pending or adjusted, resources active or queued RDY: measurement has been terminated, valid results are available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.state.clone()
```

Subgroups

7.4.1.1 All

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:STATe:ALL
```

class All

All commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class FetchStruct

Response structure. Fields:

- Main_State: enums.ResourceState: OFF | RDY | RUN OFF: measurement switched off, no resources allocated, no results available (when entered after STOP...) RDY: measurement has been terminated, valid results are available RUN: measurement running (after INITiate..., READ...) , synchronization pending or adjusted, resources active or queued
- Sync_State: enums.ResourceState: PEND | ADJ | INV PEND: waiting for resource allocation, adjustment, hardware switching ('pending') ADJ: all necessary adjustments finished, measurement running ('adjusted') INV: not applicable because MainState: OFF or RDY ('invalid')
- Resource_State: enums.ResourceState: QUE | ACT | INV QUE: measurement without resources, no results available ('queued') ACT: resources allocated, acquisition of results in progress but not complete ('active') INV: not applicable because MainState: OFF or RDY ('invalid')

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:STATe:ALL
value: FetchStruct = driver.multiEval.state.all.fetch()
```

Queries the main measurement state and the measurement substates. Both measurement substates are relevant for running measurements only. Use FETCh:...:STATe? to query the main measurement state only. Use INITiate..., STOP..., ABORT... to change the measurement state.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.2 Trace

class Trace

Trace commands group definition. 178 total commands, 12 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.clone()
```

Subgroups

7.4.2.1 EvMagnitude

class EvMagnitude

EvMagnitude commands group definition. 6 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.evMagnitude.clone()
```

Subgroups

7.4.2.1.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:CURRENT
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:CURRENT
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:CURRENT
value: List[float] = driver.multiEval.trace.evMagnitude.current.fetch()
```

Returns the values of the RMS EVM traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_evm: float Range: 0 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:CURRent
value: List[float] = driver.multiEval.trace.evMagnitude.current.read()
```

Returns the values of the RMS EVM traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_evm: float Range: 0 % to 100 %, Unit: %

7.4.2.1.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:AVERage
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:AVERage
value: List[float] = driver.multiEval.trace.evMagnitude.average.fetch()
```

Returns the values of the RMS EVM traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_evm: float Range: 0 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:AVERage
value: List[float] = driver.multiEval.trace.evMagnitude.average.read()
```

Returns the values of the RMS EVM traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_evm: float Range: 0 % to 100 %, Unit: %

7.4.2.1.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:EVMagnitude:MAXimum
value: List[float] = driver.multiEval.trace.evMagnitude.maximum.fetch()
```

Returns the values of the RMS EVM traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_evm: float Range: 0 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:EVMagnitude:MAXimum
value: List[float] = driver.multiEval.trace.evMagnitude.maximum.read()
```

Returns the values of the RMS EVM traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_evm: float Range: 0 % to 100 %, Unit: %

7.4.2.2 Merror

class Merror

Merror commands group definition. 6 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.merror.clone()
```

Subgroups

7.4.2.2.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:MERRor:CURRent
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:MERRor:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class FetchStruct

Response structure. Fields:

- Relaibiltiy: int: No parameter help available
- Current_Merr: List[float]: No parameter help available

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:MERRor:CURRent
value: FetchStruct = driver.multiEval.trace.merror.current.fetch()
```

Returns the values of the RMS magnitude error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

return structure: for return value, see the help for FetchStruct structure arguments.

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:MERRor:CURRent
value: List[float] = driver.multiEval.trace.merror.current.read()
```

Returns the values of the RMS magnitude error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_merr: float Range: -100 % to 100 %, Unit: %

7.4.2.2.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:MERRor:AVERage
FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:MERRor:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:MERRor:AVERage
value: List[float] = driver.multiEval.trace.merror.average.fetch()
```

Returns the values of the RMS magnitude error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_merr: float Range: -100 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:MERRor:AVERage
value: List[float] = driver.multiEval.trace.merror.average.read()
```

Returns the values of the RMS magnitude error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_merr: float Range: -100 % to 100 %, Unit: %

7.4.2.2.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:MERRor:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:MERRor:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:MERRor:MAXimum
value: List[float] = driver.multiEval.trace.merror.maximum.fetch()
```

Returns the values of the RMS magnitude error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_merr: float Range: -100 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:MERRor:MAXimum
value: List[float] = driver.multiEval.trace.merror.maximum.read()
```

Returns the values of the RMS magnitude error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_merr: float Range: -100 % to 100 %, Unit: %

7.4.2.3 Perror

class Perror

Perror commands group definition. 6 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.perror.clone()
```


Subgroups

7.4.2.3.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:PERRor:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:PERRor:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:PERRor:CURRent
value: List[float] = driver.multiEval.trace.perror.current.fetch()
```

Returns the values of the RMS phase error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_perr: float Range: -180 deg to 180 deg, Unit: deg

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:PERRor:CURRent
value: List[float] = driver.multiEval.trace.perror.current.read()
```

Returns the values of the RMS phase error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_perr: float Range: -180 deg to 180 deg, Unit: deg

7.4.2.3.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:PERRor:AVERage
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:PERRor:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:PERRor:AVERage
value: List[float] = driver.multiEval.trace.perror.average.fetch()
```

Returns the values of the RMS phase error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_perr: float Range: -180 deg to 180 deg, Unit: deg

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:PERRor:AVERage
value: List[float] = driver.multiEval.trace.perror.average.read()
```

Returns the values of the RMS phase error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_perr: float Range: -180 deg to 180 deg, Unit: deg

7.4.2.3.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:PERRor:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:PERRor:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:PERRor:MAXimum
value: List[float] = driver.multiEval.trace.perror.maximum.fetch()
```

Returns the values of the RMS phase error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_perr: float Range: -180 deg to 180 deg, Unit: deg

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:PERRor:MAXimum
value: List[float] = driver.multiEval.trace.perror.maximum.read()
```

Returns the values of the RMS phase error traces. The values cover a time interval of 500 s and contain one value per chip. The results of the current, average and maximum traces can be retrieved.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_perr: float Range: -180 deg to 180 deg, Unit: deg

7.4.2.4 Acp

class Acp

Acp commands group definition. 36 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.acp.clone()
```

Subgroups

7.4.2.4.1 Current

class Current

Current commands group definition. 6 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.acp.current.clone()
```

Subgroups

7.4.2.4.1.1 Relative

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:CURRent:RELative
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:CURRent:RELative
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:CURRent:RELative
```

class Relative

Relative commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:CURRent[:RELative]
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.current.relative.
↳calculate()
```

Returns the relative adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position

within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:CURRent[:RELative]
value: List[float] = driver.multiEval.trace.acp.current.relative.fetch()
```

Returns the relative adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:CURRent[:RELative]
value: List[float] = driver.multiEval.trace.acp.current.relative.read()
```

Returns the relative adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_acp: No help available

7.4.2.4.1.2 Absolute

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:CURRent:ABSolute
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:CURRent:ABSolute
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:CURRent:ABSolute
```

class Absolute

Absolute commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:TRACe:ACP:CURRENT:ABSolute
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.current.absolute.
↳calculate()
```

Returns the absolute adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method `RsCmwCdma2kMeas.Configure.MultiEval.Acp.Offsets`. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by `FETCh` and `READ` commands. `CALCulate` commands return limit check results instead, one value for each result listed below.

Use `RsCmwCdma2kMeas.reliability.last_value` to read the updated reliability indicator.

return current_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:ACP:CURRENT:ABSolute
value: List[float] = driver.multiEval.trace.acp.current.absolute.fetch()
```

Returns the absolute adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method `RsCmwCdma2kMeas.Configure.MultiEval.Acp.Offsets`. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by `FETCh` and `READ` commands. `CALCulate` commands return limit check results instead, one value for each result listed below.

Use `RsCmwCdma2kMeas.reliability.last_value` to read the updated reliability indicator.

return current_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:ACP:CURRENT:ABSolute
value: List[float] = driver.multiEval.trace.acp.current.absolute.read()
```

Returns the absolute adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method `RsCmwCdma2kMeas.Configure.MultiEval.Acp.Offsets`. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by `FETCh` and `READ` commands. `CALCulate` commands return limit check results instead, one value for each result listed below.

Use `RsCmwCdma2kMeas.reliability.last_value` to read the updated reliability indicator.

return current_acp: No help available

7.4.2.4.2 Extended

class Extended

Extended commands group definition. 18 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.acp.extended.clone()
```

Subgroups

7.4.2.4.2.1 Current

class Current

Current commands group definition. 6 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.acp.extended.current.clone()
```

Subgroups

7.4.2.4.2.2 Relative

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:CURRent:RELative
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:CURRent:RELative
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:CURRent:RELative
```

class Relative

Relative commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:EXTended:CURRent[:RELative]
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.extended.current.
↪relative.calculate()
```

Returns the adjacent channel relative power measured in dBc at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position

within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_acp: No help available

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:EXTended:CURRent[:RELative]
value: List[float] = driver.multiEval.trace.acp.extended.current.relative.
↪fetch()
```

Returns the adjacent channel relative power measured in dBc at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:EXTended:CURRent[:RELative]
value: List[float] = driver.multiEval.trace.acp.extended.current.relative.read()
```

Returns the adjacent channel relative power measured in dBc at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_acp: No help available

7.4.2.4.2.3 Absolute

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:CURRent:ABSolute
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:CURRent:ABSolute
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:CURRent:ABSolute
```

class Absolute

Absolute commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳ :MEValuation:TRACe:ACP:EXTended:CURRent:ABSolute
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.extended.current.
↳ absolute.calculate()
```

Returns the adjacent channel absolute power measured in dBm at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↳ :MEValuation:TRACe:ACP:EXTended:CURRent:ABSolute
value: List[float] = driver.multiEval.trace.acp.extended.current.absolute.
↳ fetch()
```

Returns the adjacent channel absolute power measured in dBm at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>
↳ :MEValuation:TRACe:ACP:EXTended:CURRent:ABSolute
value: List[float] = driver.multiEval.trace.acp.extended.current.absolute.read()
```

Returns the adjacent channel absolute power measured in dBm at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return current_acp: No help available

7.4.2.4.2.4 Average

class Average

Average commands group definition. 6 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.acp.extended.average.clone()
```

Subgroups

7.4.2.4.2.5 Relative

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:AVERage:RELative
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:AVERage:RELative
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:AVERage:RELative
```

class Relative

Relative commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:EXTended:AVERage[:RELative]
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.extended.average.
↳relative.calculate()
```

Returns the adjacent channel relative power measured in dBc at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:EXTended:AVERage[:RELative]
value: List[float] = driver.multiEval.trace.acp.extended.average.relative.
↳fetch()
```

Returns the adjacent channel relative power measured in dBc at a series of frequencies. The frequencies are determined by the offset values defined via the command method

RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status) . The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:EXTended:AVERage[:RELative]
value: List[float] = driver.multiEval.trace.acp.extended.average.relative.read()
```

Returns the adjacent channel relative power measured in dBc at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status) . The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

7.4.2.4.2.6 Absolute

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:AVERage:ABSolute
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:AVERage:ABSolute
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:AVERage:ABSolute
```

class Absolute

Absolute commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:EXTended:AVERage:ABSolute
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.extended.average.
↳absolute.calculate()
```

Returns the adjacent channel absolute power measured in dBm at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status) . The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:EXTended:AVERage:ABSolute
value: List[float] = driver.multiEval.trace.acp.extended.average.absolute.
↪fetch()
```

Returns the adjacent channel absolute power measured in dBm at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:EXTended:AVERage:ABSolute
value: List[float] = driver.multiEval.trace.acp.extended.average.absolute.read()
```

Returns the adjacent channel absolute power measured in dBm at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

7.4.2.4.2.7 Maximum

class Maximum

Maximum commands group definition. 6 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.acp.extended.maximum.clone()
```

Subgroups

7.4.2.4.2.8 Relative

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:MAXimum:RELative
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:MAXimum:RELative
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:MAXimum:RELative
```

class Relative

Relative commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:EXTended:MAXimum[:RELative]
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.extended.maximum.
↳relative.calculate()
```

Returns the adjacent channel relative power measured in dBc at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:EXTended:MAXimum[:RELative]
value: List[float] = driver.multiEval.trace.acp.extended.maximum.relative.
↳fetch()
```

Returns the adjacent channel relative power measured in dBc at a series of frequencies. The frequencies are determined by the offset values defined via the command method

RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status) . The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:EXTended:MAXimum[:RELative]
value: List[float] = driver.multiEval.trace.acp.extended.maximum.relative.read()
```

Returns the adjacent channel relative power measured in dBc at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status) . The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

7.4.2.4.2.9 Absolute

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:MAXimum:ABSolute
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:MAXimum:ABSolute
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:EXTended:MAXimum:ABSolute
```

class Absolute

Absolute commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:EXTended:MAXimum:ABSolute
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.extended.maximum.
↪absolute.calculate()
```

Returns the adjacent channel absolute power measured in dBm at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status) . The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:EXTended:MAXimum:ABSolute
value: List[float] = driver.multiEval.trace.acp.extended.maximum.absolute.
↪fetch()
```

Returns the adjacent channel absolute power measured in dBm at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:EXTended:MAXimum:ABSolute
value: List[float] = driver.multiEval.trace.acp.extended.maximum.absolute.read()
```

Returns the adjacent channel absolute power measured in dBm at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.Extended.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

7.4.2.4.3 Average

class Average

Average commands group definition. 6 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.acp.average.clone()
```

Subgroups

7.4.2.4.3.1 Relative

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:AVERage:RELative
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:AVERage:RELative
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:AVERage:RELative
```

class Relative

Relative commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:AVERage[:RELative]
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.average.relative.
↳calculate()
```

Returns the relative adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:AVERage[:RELative]
value: List[float] = driver.multiEval.trace.acp.average.relative.fetch()
```

Returns the relative adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method

RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets. All defined offset values are considered (irrespective of their activation status) . The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:AVERage[:RELative]
value: List[float] = driver.multiEval.trace.acp.average.relative.read()
```

Returns the relative adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets. All defined offset values are considered (irrespective of their activation status) . The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

7.4.2.4.3.2 Absolute

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:AVERage:ABSolute
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:AVERage:ABSolute
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:AVERage:ABSolute
```

class Absolute

Absolute commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:ACP:AVERage:ABSolute
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.average.absolute.
↪calculate()
```

Returns the absolute adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets. All defined offset values are considered (irrespective of their activation status) . The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:ACP:AVERage:ABSolute
value: List[float] = driver.multiEval.trace.acp.average.absolute.fetch()
```

Returns the absolute adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:ACP:AVERage:ABSolute
value: List[float] = driver.multiEval.trace.acp.average.absolute.read()
```

Returns the absolute adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return average_acp: No help available

7.4.2.4.4 Maximum

class Maximum

Maximum commands group definition. 6 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.acp.maximum.clone()
```

Subgroups

7.4.2.4.4.1 Relative

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:MAXimum:RELative
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:MAXimum:RELative
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:MAXimum:RELative
```

class Relative

Relative commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:MAXimum[:RELative]
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.maximum.relative.
↳calculate()
```

Returns the relative adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:MAXimum[:RELative]
value: List[float] = driver.multiEval.trace.acp.maximum.relative.fetch()
```

Returns the relative adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:MAXimum[:RELative]
value: List[float] = driver.multiEval.trace.acp.maximum.relative.read()
```

Returns the relative adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method `RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets`. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by `FETCH` and `READ` commands. `CALCulate` commands return limit check results instead, one value for each result listed below.

Use `RsCmwCdma2kMeas.reliability.last_value` to read the updated reliability indicator.

return maximum_acp: No help available

7.4.2.4.4.2 Absolute

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:MAXimum:ABSolute
FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:MAXimum:ABSolute
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:MAXimum:ABSolute
```

class Absolute

Absolute commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:ACP:MAXimum:ABSolute
value: List[enums.ResultStatus2] = driver.multiEval.trace.acp.maximum.absolute.
↳calculate()
```

Returns the absolute adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method `RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets`. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by `FETCH` and `READ` commands. `CALCulate` commands return limit check results instead, one value for each result listed below.

Use `RsCmwCdma2kMeas.reliability.last_value` to read the updated reliability indicator.

return maximum_acp: No help available

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:MAXimum:ABSolute
value: List[float] = driver.multiEval.trace.acp.maximum.absolute.fetch()
```

Returns the absolute adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method `RsCmwCdma2kMeas.Configure.MultiEval.Acp.foffsets`. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by `FETCH` and `READ` commands. `CALCulate` commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:ACP:MAXimum:ABSolute
value: List[float] = driver.multiEval.trace.acp.maximum.absolute.read()
```

Returns the absolute adjacent channel power measured at a series of frequencies. The frequencies are determined by the offset values defined via the command method RsCmwCdma2kMeas.Configure.MultiEval.Acp.offsets. All defined offset values are considered (irrespective of their activation status). The current, average and maximum traces can be retrieved. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return maximum_acp: No help available

7.4.2.5 Obw

class Obw

Obw commands group definition. 3 total commands, 1 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.obw.clone()
```

Subgroups

7.4.2.5.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:OBW:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:OBW:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:OBW:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:OBW:CURRent
value: List[enums.ResultStatus2] = driver.multiEval.trace.obw.current.
↪ calculate()
```

Returns the traces of the current occupied bandwidth (OBW) . The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return curr_obw: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:OBW:CURRent
value: List[float] = driver.multiEval.trace.obw.current.fetch()
```

Returns the traces of the current occupied bandwidth (OBW) . The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return curr_obw: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:OBW:CURRENT
value: List[float] = driver.multiEval.trace.obw.current.read()
```

Returns the traces of the current occupied bandwidth (OBW) . The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return curr_obw: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz

7.4.2.6 Spectrum

class Spectrum

Spectrum commands group definition. 3 total commands, 1 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.spectrum.clone()
```

Subgroups

7.4.2.6.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:SPECtrum:CURRent
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:SPECtrum:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:SPECtrum:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:SPECtrum:CURRent
value: List[enums.ResultStatus2] = driver.multiEval.trace.spectrum.current.
↪ calculate()
```

Returns the power spectrum traces across the frequency span (8 MHz) . The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return curr_spectrum: float Frequency-dependent power values. Each of the traces contains 1667 values. Range: -100 dB to +57 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:SPECtrum:CURRent
value: List[float] = driver.multiEval.trace.spectrum.current.fetch()
```

Returns the power spectrum traces across the frequency span (8 MHz) . The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return curr_spectrum: float Frequency-dependent power values. Each of the traces contains 1667 values. Range: -100 dB to +57 dB , Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:SPECtrum:CURRent
value: List[float] = driver.multiEval.trace.spectrum.current.read()
```

Returns the power spectrum traces across the frequency span (8 MHz) . The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return curr_spectrum: float Frequency-dependent power values. Each of the traces contains 1667 values. Range: -100 dB to +57 dB , Unit: dB

7.4.2.7 Cdp

class Cdp

Cdp commands group definition. 28 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cdp.clone()
```

Subgroups

7.4.2.7.1 Isignal

class Isignal

Isignal commands group definition. 14 total commands, 6 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cdp.isignal.clone()
```

Subgroups

7.4.2.7.1.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISIGnal:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISIGnal:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISIGnal:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CDP:ISIGnal:CURRent
value: List[enums.ResultStatus2] = driver.multiEval.trace.cdp.isignal.current.
↪calculate()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_current: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:CURRent
value: List[float] = driver.multiEval.trace.cdp.isignal.current.fetch()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_current: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:CURRent
value: List[float] = driver.multiEval.trace.cdp.isignal.current.read()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_curr: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.7.1.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:AVERage
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:TRACe:CDP:ISIGnal:AVERage
value: List[enums.ResultStatus2] = driver.multiEval.trace.cdp.isignal.average.
↪calculate()
```


Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_average: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:AVERage
value: List[float] = driver.multiEval.trace.cdp.isignal.average.fetch()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_average: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:AVERage
value: List[float] = driver.multiEval.trace.cdp.isignal.average.read()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_aver: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.7.1.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:TRACe:CDP:ISIGnal:MAXimum
value: List[enums.ResultStatus2] = driver.multiEval.trace.cdp.isignal.maximum.
↪calculate()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:MAXimum
value: List[float] = driver.multiEval.trace.cdp.isignal.maximum.fetch()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:ISIGnal:MAXimum
value: List[float] = driver.multiEval.trace.cdp.isignal.maximum.read()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.7.1.4 Minimum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISIGnal:MINimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISIGnal:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISIGnal:MINimum
```

class Minimum

Minimum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
→ :MEvaluation:TRACe:CDP:ISIGnal:MINimum
value: List[enums.ResultStatus2] = driver.multiEval.trace.cdp.isignal.minimum.
→ calculate()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_min: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISIGnal:MINimum
value: List[float] = driver.multiEval.trace.cdp.isignal.minimum.fetch()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_min: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISIGnal:MINimum
value: List[float] = driver.multiEval.trace.cdp.isignal.minimum.read()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_min: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.7.1.5 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISignal:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateB]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISignal:STATe
value: List[enums.SigChStateB] = driver.multiEval.trace.cdp.isignal.state.
↪ fetch()
```

Return the states of the code domain power (CDP) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_ch_state: INVisible | ACTive | IACTive The number of results depends on the selected spreading factor: SF=16, 32, 64. INV: No channel available ACTive: Active code channel IACTive: Inactive code channel

7.4.2.7.1.6 Limit

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISignal:LIMit
```

class Limit

Limit commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:ISignal:LIMit
value: List[float] = driver.multiEval.trace.cdp.isignal.limit.fetch()
```

Return limit check results for the code domain power (CDP) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return isig_limit: float Return the exceeded limits as float values. The number of results depends on the selected spreading factor: SF=16, 32, 64.

7.4.2.7.2 Qsignal

class Qsignal

Qsignal commands group definition. 14 total commands, 6 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cdp.qsignal.clone()
```

Subgroups

7.4.2.7.2.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
→:MEvaluation:TRACe:CDP:QSIGnal:CURRent
value: List[enums.ResultStatus2] = driver.multiEval.trace.cdp.qsignal.current.
→calculate()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_current: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:CURRent
value: List[float] = driver.multiEval.trace.cdp.qsignal.current.fetch()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_current: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSiGnal:CURRent
value: List[float] = driver.multiEval.trace.cdp.qsignal.current.read()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_curr: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.7.2.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSiGnal:AVERage
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSiGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSiGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CDP:QSiGnal:AVERage
value: List[enums.ResultStatus2] = driver.multiEval.trace.cdp.qsignal.average.
↪calculate()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_average: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSiGnal:AVERage
value: List[float] = driver.multiEval.trace.cdp.qsignal.average.fetch()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_average: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:AVERage
value: List[float] = driver.multiEval.trace.cdp.qsignal.average.read()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_aver: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.7.2.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CDP:QSIGnal:MAXimum
value: List[enums.ResultStatus2] = driver.multiEval.trace.cdp.qsignal.maximum.
↪calculate()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:QSIGnal:MAXimum
value: List[float] = driver.multiEval.trace.cdp.qsignal.maximum.fetch()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:QSIGnal:MAXimum
value: List[float] = driver.multiEval.trace.cdp.qsignal.maximum.read()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.7.2.4 Minimum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:QSIGnal:MINimum
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:QSIGnal:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:QSIGnal:MINimum
```

class Minimum

Minimum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:TRACe:CDP:QSIGnal:MINimum
value: List[enums.ResultStatus2] = driver.multiEval.trace.cdp.qsignal.minimum.
↳calculate()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by

FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_min: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:QSiGnal:MINimum
value: List[float] = driver.multiEval.trace.cdp.qsignal.minimum.fetch()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_min: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:QSiGnal:MINimum
value: List[float] = driver.multiEval.trace.cdp.qsignal.minimum.read()
```

Returns the values of the code domain power (CDP) I-Signal and Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_min: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.7.2.5 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:QSiGnal:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateB]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDP:QSiGnal:STATE
value: List[enums.SigChStateB] = driver.multiEval.trace.cdp.qsignal.state.
fetch()
```

(continues on next page)

(continued from previous page)

Return the states of the code domain power (CDP) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_qsig_ch_state: INVisible | ACTive | IACTive The number of results depends on the selected spreading factor: SF=16, 32, 64. INV: No channel available ACTive: Active code channel IACTive: Inactive code channel

7.4.2.7.2.6 Limit

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:LIMit
```

class Limit

Limit commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDP:QSIGnal:LIMit
value: List[float] = driver.multiEval.trace.cdp.qsignal.limit.fetch()
```

Return limit check results for the code domain power (CDP) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return qsig_limit: float Return the exceeded limits as float values. The number of results depends on the selected spreading factor: SF=16, 32, 64.

7.4.2.8 Cde

class Cde

Cde commands group definition. 22 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cde.clone()
```

Subgroups

7.4.2.8.1 Isignal

class Isignal

Isignal commands group definition. 11 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cde.isignal.clone()
```

Subgroups

7.4.2.8.1.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:CDE:ISIGnal:CURRent
value: List[enums.ResultStatus2] = driver.multiEval.trace.cde.isignal.current.
↳calculate()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_curr: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:CURRent
value: List[float] = driver.multiEval.trace.cde.isignal.current.fetch()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdp_isig_curr: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:ISIGnal:CURRent
value: List[float] = driver.multiEval.trace.cde.isignal.current.read()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_isig_curr: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.8.1.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:ISIGnal:AVERage
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:ISIGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:ISIGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:TRACe:CDE:ISIGnal:AVERage
value: List[enums.ResultStatus2] = driver.multiEval.trace.cde.isignal.average.
↪calculate()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_isig_average: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:ISIGnal:AVERage
value: List[float] = driver.multiEval.trace.cde.isignal.average.fetch()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_isig_average: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:AVERage
value: List[float] = driver.multiEval.trace.cde.isignal.average.read()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_isig_aver: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.8.1.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CDE:ISIGnal:MAXimum
value: List[enums.ResultStatus2] = driver.multiEval.trace.cde.isignal.maximum.
↪calculate()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_isig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:MAXimum
value: List[float] = driver.multiEval.trace.cde.isignal.maximum.fetch()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_isig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:MAXimum
value: List[float] = driver.multiEval.trace.cde.isignal.maximum.read()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_isig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.8.1.4 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateB]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:ISIGnal:STATE
value: List[enums.SigChStateB] = driver.multiEval.trace.cde.isignal.state.
↪ fetch()
```

Return the states of the code domain error (CDE) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cde_isig_ch_state: INVisible | ACTIVE | IACTIVE The number of results depends on the selected spreading factor: SF=16, 32, 64. INV: No channel available ACTIVE: Active code channel IACTIVE: Inactive code channel

7.4.2.8.1.5 Limit

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:ISIGnal:LIMit
```

class Limit

Limit commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:ISIGnal:LIMit
value: List[float] = driver.multiEval.trace.cde.isignal.limit.fetch()
```

Return limit check results for the code domain error (CDE) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return isig_limit: float Return the exceeded limits as float values. The number of results depends on the selected spreading factor: SF=16, 32, 64.

7.4.2.8.2 Qsignal

class Qsignal

Qsignal commands group definition. 11 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cde.qsignal.clone()
```

Subgroups

7.4.2.8.2.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSIGnal:CURRent
FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSIGnal:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSIGnal:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:TRACe:CDE:QSIGnal:CURRent
value: List[enums.ResultStatus2] = driver.multiEval.trace.cde.qsignal.current.
↪calculate()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_qsig_current: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSiGnal:CURRent
value: List[float] = driver.multiEval.trace.cde.qsignal.current.fetch()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_qsig_current: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSiGnal:CURRent
value: List[float] = driver.multiEval.trace.cde.qsignal.current.read()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_qsig_curr: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.8.2.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSiGnal:AVERage
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSiGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSiGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]


```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:TRACe:CDE:QSIGnal:AVERage
value: List[enums.ResultStatus2] = driver.multiEval.trace.cde.qsignal.average.
↳calculate()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_qsig_aver: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSIGnal:AVERage
value: List[float] = driver.multiEval.trace.cde.qsignal.average.fetch()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_qsig_average: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CDE:QSIGnal:AVERage
value: List[float] = driver.multiEval.trace.cde.qsignal.average.read()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_qsig_aver: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.8.2.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:QSIGnal:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:QSIGnal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:QSIGnal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
→ :MEvaluation:TRACe:CDE:QSIGnal:MAXimum
value: List[enums.ResultStatus2] = driver.multiEval.trace.cde.qsignal.maximum.
→ calculate()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_qsig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:QSIGnal:MAXimum
value: List[float] = driver.multiEval.trace.cde.qsignal.maximum.fetch()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_qsig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:QSIGnal:MAXimum
value: List[float] = driver.multiEval.trace.cde.qsignal.maximum.read()
```

Returns the values of the code domain error (CDE) I-Signal and Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cdep_qsig_max: float The number of results corresponds to the selected spreading factor (method RsCmwCdma2kMeas.Configure.MultiEval.sfactor) , for example 16 results for SF16. Range: -70 dB to 0 dB, Unit: dB

7.4.2.8.2.4 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:QSiGnal:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateB]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:QSiGnal:STATe
value: List[enums.SigChStateB] = driver.multiEval.trace.cde.qsignal.state.
↪ fetch()
```

Return the states of the code domain error (CDE) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cde_qsig_ch_state: INVisible | ACTive | IACTive The number of results depends on the selected spreading factor: SF=16, 32, 64. INV: No channel available ACTive: Active code channel IACTive: Inactive code channel

7.4.2.8.2.5 Limit

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:QSiGnal:LIMit
```

class Limit

Limit commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CDE:QSiGnal:LIMit
value: List[float] = driver.multiEval.trace.cde.qsignal.limit.fetch()
```

Return limit check results for the code domain error (CDE) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return qsig_limit: float Return the exceeded limits as float values. The number of results depends on the selected spreading factor: SF=16, 32, 64.

7.4.2.9 Cp

class Cp

Cp commands group definition. 26 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cp.clone()
```

Subgroups

7.4.2.9.1 Isignal

class Isignal

Isignal commands group definition. 13 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cp.isignal.clone()
```

Subgroups

7.4.2.9.1.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliabilty: float: No parameter help available
- Pich: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Dcch: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Sch_1: float: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'

- Pich: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Dcch: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Sch_1: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CP:ISIGnal:CURRent
value: CalculateStruct = driver.multiEval.trace.cp.isignal.current.calculate()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:CURRent
value: ResultData = driver.multiEval.trace.cp.isignal.current.fetch()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:CURRent
value: ResultData = driver.multiEval.trace.cp.isignal.current.read()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.9.1.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:AVERage
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: float: No parameter help available
- Pich: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Dcch: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Sch_1: float: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Dcch: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Sch_1: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:CP:ISIGnal:AVERage
value: CalculateStruct = driver.multiEval.trace.cp.isignal.average.calculate()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:AVERage
value: ResultData = driver.multiEval.trace.cp.isignal.average.fetch()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:AVERage
value: ResultData = driver.multiEval.trace.cp.isignal.average.read()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.9.1.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliabilty: float: No parameter help available
- Pich: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Dcch: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Sch_1: float: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Dcch: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Sch_1: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:CP:ISIGnal:MAXimum
value: CalculateStruct = driver.multiEval.trace.cp.isignal.maximum.calculate()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:MAXimum
value: ResultData = driver.multiEval.trace.cp.isignal.maximum.fetch()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISignal:MAXimum
value: ResultData = driver.multiEval.trace.cp.isignal.maximum.read()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.9.1.4 Minimum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISignal:MINimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISignal:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISignal:MINimum
```

class Minimum

Minimum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliabilty: float: No parameter help available
- Pich: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Dcch: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Sch_1: float: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Dcch: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB
- Sch_1: float: float RMS channel power values for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -60 dB to 0 dB , Unit: dB

calculate() → CalculateStruct


```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:CP:ISIGnal:MINimum
value: CalculateStruct = driver.multiEval.trace.cp.isignal.minimum.calculate()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:MINimum
value: ResultData = driver.multiEval.trace.cp.isignal.minimum.fetch()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:MINimum
value: ResultData = driver.multiEval.trace.cp.isignal.minimum.read()
```

Returns the values of the channel power I-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.9.1.5 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:ISIGnal:STATe
value: List[enums.SigChStateA] = driver.multiEval.trace.cp.isignal.state.fetch()
```

Return the states of the channel power (CP) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cp_isig_ch_state: No help available

7.4.2.9.2 Qsignal

class Qsignal

Qsignal commands group definition. 13 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cp.qsignal.clone()
```

Subgroups

7.4.2.9.2.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSIGnal:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSIGnal:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSIGnal:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: float: decimal 'Reliability Indicator'
- Fch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Sch_0: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Each: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Ccch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Sch_0: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

- Each: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Ccch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CP:QSiGnal:CURRent
value: CalculateStruct = driver.multiEval.trace.cp.qsignal.current.calculate()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSiGnal:CURRent
value: ResultData = driver.multiEval.trace.cp.qsignal.current.fetch()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSiGnal:CURRent
value: ResultData = driver.multiEval.trace.cp.qsignal.current.read()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.9.2.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSiGnal:AVERage
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSiGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSiGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: float: decimal 'Reliability Indicator'
- Fch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Sch_0: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Each: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Ccch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Sch_0: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Each: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Ccch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>  
↳:MEvaluation:TRACe:CP:QSIGnal:AVERage  
value: CalculateStruct = driver.multiEval.trace.cp.qsignal.average.calculate()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSIGnal:AVERage  
value: ResultData = driver.multiEval.trace.cp.qsignal.average.fetch()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSiGnal:AVErage
value: ResultData = driver.multiEval.trace.cp.qsignal.average.read()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.9.2.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSiGnal:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSiGnal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSiGnal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliabilty: float: decimal 'Reliability Indicator'
- Fch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Sch_0: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Each: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Ccch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

- Sch_0: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Each: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Ccch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CP:QSignal:MAXimum
value: CalculateStruct = driver.multiEval.trace.cp.qsignal.maximum.calculate()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSignal:MAXimum
value: ResultData = driver.multiEval.trace.cp.qsignal.maximum.fetch()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSignal:MAXimum
value: ResultData = driver.multiEval.trace.cp.qsignal.maximum.read()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.9.2.4 Minimum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSIGnal:MINimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSIGnal:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP:QSIGnal:MINimum
```

class Minimum

Minimum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: float: decimal 'Reliability Indicator'
- Fch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Sch_0: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Each: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Ccch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Sch_0: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Each: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB
- Ccch: float: float RMS channel power values for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: -60 dB to 0 dB , Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CP:QSIGnal:MINimum
value: CalculateStruct = driver.multiEval.trace.cp.qsignal.minimum.calculate()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP:QSiGnal:MINimum
value: ResultData = driver.multiEval.trace.cp.qsignal.minimum.fetch()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP:QSiGnal:MINimum
value: ResultData = driver.multiEval.trace.cp.qsignal.minimum.read()
```

Returns the values of the channel power Q-Signal traces. The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.9.2.5 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP:QSiGnal:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP:QSiGnal:STATE
value: List[enums.SigChStateA] = driver.multiEval.trace.cp.qsignal.state.fetch()
```

Return the states of the channel power (CP) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cp_qsig_ch_state: No help available

7.4.2.10 Cpo

class Cpo

Cpo commands group definition. 20 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cpo.clone()
```

Subgroups

7.4.2.10.1 Isignal

class Isignal

Isignal commands group definition. 10 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cpo.isignal.clone()
```

Subgroups

7.4.2.10.1.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:ISignal:CURRENT
FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:ISignal:CURRENT
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:ISignal:CURRENT
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: enums.ResultStatus2: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Dcch: enums.ResultStatus2: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Sch_1: enums.ResultStatus2: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: float: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: $- \cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Dcch: float: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: $- \cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Sch_1: float: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: $- \cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:TRACe:CPO:ISIGnal:CURRent
value: CalculateStruct = driver.multiEval.trace.cpo.isignal.current.calculate()
```

Returns the phase offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CPO:ISIGnal:CURRent
value: ResultData = driver.multiEval.trace.cpo.isignal.current.fetch()
```

Returns the phase offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CPO:ISIGnal:CURRent
value: ResultData = driver.multiEval.trace.cpo.isignal.current.read()
```

Returns the phase offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.10.1.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:ISIGnal:AVERage
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:ISIGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:ISIGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: enums.ResultStatus2: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Dcch: enums.ResultStatus2: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Sch_1: enums.ResultStatus2: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: float: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Dcch: float: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Sch_1: float: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:TRACe:CP0:ISIGnal:AVERage
value: CalculateStruct = driver.multiEval.trace.cpo.isignal.average.calculate()
```

Returns the phase offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:ISIGnal:AVERage
value: ResultData = driver.multiEval.trace.cpo.isignal.average.fetch()
```

Returns the phase offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:ISignal:AVERage
value: ResultData = driver.multiEval.trace.cpo.isignal.average.read()
```

Returns the phase offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.10.1.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:ISignal:MAXimum
FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:ISignal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:ISignal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: enums.ResultStatus2: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Dcch: enums.ResultStatus2: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Sch_1: enums.ResultStatus2: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: float: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Dcch: float: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Sch_1: float: float Phase offset for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:CPO:ISIGnal:MAXimum
value: CalculateStruct = driver.multiEval.trace.cpo.isignal.maximum.calculate()
```

Returns the phase offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:ISIGnal:MAXimum
value: ResultData = driver.multiEval.trace.cpo.isignal.maximum.fetch()
```

Returns the phase offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:ISIGnal:MAXimum
value: ResultData = driver.multiEval.trace.cpo.isignal.maximum.read()
```

Returns the phase offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average, maximum and minimum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.10.1.4 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:ISIGnal:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:ISIGnal:STATE
value: List[enums.SigChStateA] = driver.multiEval.trace.cpo.isignal.state.
↳fetch()
```

Return the states of the channel phase offset (CPO) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cpo_isig_ch_state: No help available

7.4.2.10.2 Qsignal

class Qsignal

Qsignal commands group definition. 10 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cpo.qsignal.clone()
```

Subgroups

7.4.2.10.2.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP0:QSiGnal:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP0:QSiGnal:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP0:QSiGnal:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Sch_0: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Each: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Ccch: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'

- Fch: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Sch_0: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Each: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Ccch: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CP0:QSignal:CURRent
value: CalculateStruct = driver.multiEval.trace.cpo.qsignal.current.calculate()
```

Returns the phase offset for the indicated channels in the quadrature-phase signal path (Q-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP0:QSignal:CURRent
value: ResultData = driver.multiEval.trace.cpo.qsignal.current.fetch()
```

Returns the phase offset for the indicated channels in the quadrature-phase signal path (Q-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CP0:QSignal:CURRent
value: ResultData = driver.multiEval.trace.cpo.qsignal.current.read()
```

Returns the phase offset for the indicated channels in the quadrature-phase signal path (Q-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.10.2.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:QSIGnal:AVERage
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:QSIGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:QSIGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Sch_0: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Each: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Ccch: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Sch_0: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Each: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Ccch: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:TRACe:CP0:QSIGnal:AVERage
value: CalculateStruct = driver.multiEval.trace.cpo.qsignal.average.calculate()
```


Returns the phase offset for the indicated channels in the quadrature-phase signal path (Q-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:QSiGnal:AVERage
value: ResultData = driver.multiEval.trace.cpo.qsignal.average.fetch()
```

Returns the phase offset for the indicated channels in the quadrature-phase signal path (Q-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:QSiGnal:AVERage
value: ResultData = driver.multiEval.trace.cpo.qsignal.average.read()
```

Returns the phase offset for the indicated channels in the quadrature-phase signal path (Q-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.10.2.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:QSiGnal:MAXimum
FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:QSiGnal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CP0:QSiGnal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad
- Sch_0: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - · 10E+3 mRad to · 10E+3 mRad , Unit: mRad

- Each: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Ccch: enums.ResultStatus2: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Sch_0: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Each: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad
- Ccch: float: float Reverse fundamental channel Phase offset for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - $\cdot 10E+3$ mRad to $\cdot 10E+3$ mRad , Unit: mRad

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:CPO:QSIGnal:MAXimum
value: CalculateStruct = driver.multiEval.trace.cpo.qsignal.maximum.calculate()
```

Returns the phase offset for the indicated channels in the quadrature-phase signal path (Q-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:QSIGnal:MAXimum
value: ResultData = driver.multiEval.trace.cpo.qsignal.maximum.fetch()
```

Returns the phase offset for the indicated channels in the quadrature-phase signal path (Q-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:QSIGnal:MAXimum
value: ResultData = driver.multiEval.trace.cpo.qsignal.maximum.read()
```

Returns the phase offset for the indicated channels in the quadrature-phase signal path (Q-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.10.2.4 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:QSIGnal:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CPO:QSIGnal:STATe
value: List[enums.SigChStateA] = driver.multiEval.trace.cpo.qsignal.state.
↪ fetch()
```

Return the states of the channel phase offset (CPO) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cpo_qsig_ch_state: No help available

7.4.2.11 Cto

class Cto

Cto commands group definition. 20 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cto.clone()
```

Subgroups

7.4.2.11.1 Isignal

class Isignal

Isignal commands group definition. 10 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cto.isignal.clone()
```

Subgroups

7.4.2.11.1.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:ISIGnal:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:ISIGnal:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:ISIGnal:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: enums.ResultStatus2: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Dcch: enums.ResultStatus2: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Sch_1: enums.ResultStatus2: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: float: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Dcch: float: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Sch_1: float: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:TRACe:CTO:ISIGnal:CURRENT
value: CalculateStruct = driver.multiEval.trace.cto.isignal.current.calculate()
```

Returns channel time offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CTO:ISIGnal:CURRENT
value: ResultData = driver.multiEval.trace.cto.isignal.current.fetch()
```

Returns channel time offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CTO:ISIGnal:CURRENT
value: ResultData = driver.multiEval.trace.cto.isignal.current.read()
```

Returns channel time offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.11.1.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CTO:ISIGnal:AVERage
FETCH:CDMA:MEASurement<Instance>:MEValuation:TRACe:CTO:ISIGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CTO:ISIGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: enums.ResultStatus2: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns

- Dcch: enums.ResultStatus2: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Sch_1: enums.ResultStatus2: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: float: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Dcch: float: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Sch_1: float: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CTO:ISIGnal:AVERage
value: CalculateStruct = driver.multiEval.trace.cto.isignal.average.calculate()
```

Returns channel time offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:ISIGnal:AVERage
value: ResultData = driver.multiEval.trace.cto.isignal.average.fetch()
```

Returns channel time offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:ISIGnal:AVERage
value: ResultData = driver.multiEval.trace.cto.isignal.average.read()
```

Returns channel time offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.11.1.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:TRACe:CTO:ISIGnal:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CTO:ISIGnal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:TRACe:CTO:ISIGnal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: enums.ResultStatus2: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Dcch: enums.ResultStatus2: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Sch_1: enums.ResultStatus2: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pich: float: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Dcch: float: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns
- Sch_1: float: float Time offset in nanoseconds for reverse pilot channel, reverse dedicated control channel and reverse supplemental channel 0. Range: -50 ns to +50ns , Unit: ns

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:TRACe:CTO:ISIGnal:MAXimum
value: CalculateStruct = driver.multiEval.trace.cto.isignal.maximum.calculate()
```

Returns channel time offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:TRACe:CTO:ISIGnal:MAXimum
value: ResultData = driver.multiEval.trace.cto.isignal.maximum.fetch()
```

Returns channel time offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:ISignal:MAXimum
value: ResultData = driver.multiEval.trace.cto.isignal.maximum.read()
```

Returns channel time offset for the indicated channels in the in-phase signal path (I-signal) . The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.11.1.4 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:ISignal:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:ISignal:STATe
value: List[enums.SigChStateA] = driver.multiEval.trace.cto.isignal.state.
↪ fetch()
```

Return the states of the channel time offset (CTO) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cto_isig_ch_state: No help available

7.4.2.11.2 Qsignal

class Qsignal

Qsignal commands group definition. 10 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.cto.qsignal.clone()
```

Subgroups

7.4.2.11.2.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSIGnal:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSIGnal:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSIGnal:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Sch_0: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Each: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Ccch: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Sch_0: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Each: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns

- Ccch: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:TRACe:CTO:QSiGnal:CURRent
value: CalculateStruct = driver.multiEval.trace.cto.qsignal.current.calculate()
```

Returns the values of the channel time offset Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSiGnal:CURRent
value: ResultData = driver.multiEval.trace.cto.qsignal.current.fetch()
```

Returns the values of the channel time offset Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSiGnal:CURRent
value: ResultData = driver.multiEval.trace.cto.qsignal.current.read()
```

Returns the values of the channel time offset Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.11.2.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSiGnal:AVERage
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSiGnal:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSiGnal:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'

- Fch: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Sch_0: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Each: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Ccch: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Sch_0: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Each: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Ccch: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:TRACe:CTO:QSignal:AVERage
value: CalculateStruct = driver.multiEval.trace.cto.qsignal.average.calculate()
```

Returns the values of the channel time offset Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSignal:AVERage
value: ResultData = driver.multiEval.trace.cto.qsignal.average.fetch()
```

Returns the values of the channel time offset Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSiGnal:AVErage
value: ResultData = driver.multiEval.trace.cto.qsignal.average.read()
```

Returns the values of the channel time offset Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.11.2.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSiGnal:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSiGnal:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSiGnal:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Sch_0: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Each: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Ccch: enums.ResultStatus2: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Fch: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Sch_0: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns

- Each: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns
- Ccch: float: float Time offset in nanoseconds for reverse fundamental channel, reverse supplemental channel 0, enhanced access channel and reverse common control channel. Range: - 50 ns to + 50 ns , Unit: ns

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳ :MEvaluation:TRACe:CTO:QSIGnal:MAXimum
value: CalculateStruct = driver.multiEval.trace.cto.qsignal.maximum.calculate()
```

Returns the values of the channel time offset Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSIGnal:MAXimum
value: ResultData = driver.multiEval.trace.cto.qsignal.maximum.fetch()
```

Returns the values of the channel time offset Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSIGnal:MAXimum
value: ResultData = driver.multiEval.trace.cto.qsignal.maximum.read()
```

Returns the values of the channel time offset Q-Signal traces. The results of the current, average and maximum traces can be retrieved. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.2.11.2.4 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSIGnal:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:CTO:QSignal:STATE
value: List[enums.SigChStateA] = driver.multiEval.trace.cto.qsignal.state.
↪ fetch()
```

Return the states of the channel time offset (CTO) I-Signal and Q-Signal bar graphs.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cto_qsig_ch_state: No help available

7.4.2.12 Iq

class Iq

Iq commands group definition. 2 total commands, 1 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.trace.iq.clone()
```

Subgroups

7.4.2.12.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:TRACe:IQ:CURRENT
FETCh:CDMA:MEASurement<Instance>:MEvaluation:TRACe:IQ:CURRENT
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Iphase: List[float]: No parameter help available
- Qphase: List[float]: No parameter help available

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<instance>:MEvaluation:TRACe:IQ:CURRENT
value: ResultData = driver.multiEval.trace.iq.current.fetch()
```

Returns the results in the I/Q constellation diagram. Every fourth value corresponds to a constellation point. The other values are on the path between two constellation points.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<instance>:MEvaluation:TRACe:IQ:CURRent
value: ResultData = driver.multiEval.trace.iq.current.read()
```

Returns the results in the I/Q constellation diagram. Every fourth value corresponds to a constellation point. The other values are on the path between two constellation points.

return structure: for return value, see the help for ResultData structure arguments.

7.4.3 Modulation

class Modulation

Modulation commands group definition. 15 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.modulation.clone()
```

Subgroups

7.4.3.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:MODulation:CURRent
FETCh:CDMA:MEASurement<Instance>:MEvaluation:MODulation:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:MODulation:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- Perr_Peak: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz

- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Power: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- Perr_Peak: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Power: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

class ReadStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %

- **Merr_Rms**: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- **Merr_Peak**: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- **Perr_Rms**: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- **Perr_Peak**: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset**: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- **Iq_Imbalance**: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- **Cfreq_Error**: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- **Trans_Time_Err**: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- **Ms_Power_1_M_23**: float: No parameter help available
- **Ms_Power_Wideband**: float: No parameter help available
- **Wav_Quality**: float: float Waveform quality Range: 0 to 1
- **Wav_Qual_Max_Pow**: float: No parameter help available
- **Wav_Qual_Min_Pow**: float: No parameter help available
- **Out_Of_Tol_Count**: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:MODulation:CURRENT
value: CalculateStruct = driver.multiEval.modulation.current.calculate()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:MODulation:CURRENT
value: FetchStruct = driver.multiEval.modulation.current.fetch()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

read() → ReadStruct

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:MODulation:CURRENT
value: ReadStruct = driver.multiEval.modulation.current.read()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for ReadStruct structure arguments.

7.4.3.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:MODulation:AVERage
FETCh:CDMA:MEASurement<Instance>:MEvaluation:MODulation:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:MODulation:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- Perr_Peak: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1

- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Power: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- Perr_Peak: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- Iq_Imbalance: int: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Power: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

class ReadStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg

- **Perr_Peak:** float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset:** float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- **Iq_Imbalance:** float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- **Cfreq_Error:** float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- **Trans_Time_Err:** float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- **Ms_Power_1_M_23:** float: No parameter help available
- **Ms_Power_Wideband:** float: No parameter help available
- **Wav_Quality:** float: float Waveform quality Range: 0 to 1
- **Wav_Qual_Max_Pow:** float: No parameter help available
- **Wav_Qual_Min_Pow:** float: No parameter help available
- **Out_Of_Tol_Count:** float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:MODulation:AVERage
value: CalculateStruct = driver.multiEval.modulation.average.calculate()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:MODulation:AVERage
value: FetchStruct = driver.multiEval.modulation.average.fetch()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

read() → ReadStruct

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:MODulation:AVERage
value: ReadStruct = driver.multiEval.modulation.average.read()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for ReadStruct structure arguments.

7.4.3.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:MODulation:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEvaluation:MODulation:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:MODulation:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- Perr_Peak: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Pow: float: No parameter help available

- **Out_Of_Tol_Count**: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

class ResultData

Response structure. Fields:

- **Reliability**: int: decimal 'Reliability Indicator'
- **Evm_Rms**: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- **Evm_Peak**: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- **Merr_Rms**: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- **Merr_Peak**: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- **Perr_Rms**: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- **Perr_Peak**: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset**: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- **Iq_Imbalance**: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- **Cfreq_Error**: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- **Trans_Time_Err**: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- **Ms_Power_1_M_23**: float: No parameter help available
- **Ms_Power_Wideband**: float: No parameter help available
- **Wav_Quality**: float: float Waveform quality Range: 0 to 1
- **Wav_Qual_Max_Pow**: float: No parameter help available
- **Wav_Qual_Min_Pow**: float: No parameter help available
- **Out_Of_Tol_Count**: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:MODulation:MAXimum
value: CalculateStruct = driver.multiEval.modulation.maximum.calculate()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:MODulation:MAXimum
value: ResultData = driver.multiEval.modulation.maximum.fetch()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:MODulation:MAXimum
value: ResultData = driver.multiEval.modulation.maximum.read()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for ResultData structure arguments.

7.4.3.4 Minimum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:MODulation:MINimum
FETCh:CDMA:MEASurement<Instance>:MEValuation:MODulation:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:MODulation:MINimum
```

class Minimum

Minimum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- Perr_Peak: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg

- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Pow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- Perr_Peak: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Pow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

calculate() → CalculateStruct


```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:MODulation:MINimum
value: CalculateStruct = driver.multiEval.modulation.minimum.calculate()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:MODulation:MINimum
value: ResultData = driver.multiEval.modulation.minimum.fetch()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:MODulation:MINimum
value: ResultData = driver.multiEval.modulation.minimum.read()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for ResultData structure arguments.

7.4.3.5 StandardDev

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:MODulation:SDEviation
FETCh:CDMA:MEASurement<Instance>:MEValuation:MODulation:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEValuation:MODulation:SDEviation
```

class StandardDev

StandardDev commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- Perr_Peak: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Pow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Evm_Rms: float: float Error vector magnitude RMS value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude peak value Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value Range: 0 deg to 180 deg, Unit: deg
- Perr_Peak: float: float Phase error peak value Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB, Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB, Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz, Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs

- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Pow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limits, see 'Limits (Power and Modulation) '. Range: 0 % to 100 %, Unit: %

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:MODulation:SDEviation
value: CalculateStruct = driver.multiEval.modulation.standardDev.calculate()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:MODulation:SDEviation
value: ResultData = driver.multiEval.modulation.standardDev.fetch()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:MODulation:SDEviation
value: ResultData = driver.multiEval.modulation.standardDev.read()
```

Return the current, average, minimum, maximum and standard deviation modulation single value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for ResultData structure arguments.

7.4.4 Acp

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:ACP
FETCh:CDMA:MEASurement<Instance>:MEvaluation:ACP
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:ACP
```

class Acp

Acp commands group definition. 12 total commands, 3 Sub-groups, 3 group commands

calculate() → float

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:ACP
value: float = driver.multiEval.acp.calculate()
```

Returns the ‘Out of Tolerance’ result, i.e. the percentage of measurement intervals of the statistic count (method RsCmwCdma2kMeas.Configure.MultiEval.ScounT.spectrum) exceeding the specified limits, see ‘Limits (Spectrum)’. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return out_of_tolerance: float Range: 0 % to 100 %, Unit: %

fetch() → float

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:ACP
value: float = driver.multiEval.acp.fetch()
```

Returns the ‘Out of Tolerance’ result, i.e. the percentage of measurement intervals of the statistic count (method RsCmwCdma2kMeas.Configure.MultiEval.ScounT.spectrum) exceeding the specified limits, see ‘Limits (Spectrum)’. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return out_of_tolerance: float Range: 0 % to 100 %, Unit: %

read() → float

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:ACP
value: float = driver.multiEval.acp.read()
```

Returns the ‘Out of Tolerance’ result, i.e. the percentage of measurement intervals of the statistic count (method RsCmwCdma2kMeas.Configure.MultiEval.ScounT.spectrum) exceeding the specified limits, see ‘Limits (Spectrum)’. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return out_of_tolerance: float Range: 0 % to 100 %, Unit: %

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.acp.clone()
```

Subgroups

7.4.4.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:ACP:CURRENT
FETCh:CDMA:MEASurement<Instance>:MEvaluation:ACP:CURRENT
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:ACP:CURRENT
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal
- Narrow_Band: float: float MS power, measured with a filter bandwidth of 1.23 MHz. Range: -256 dBm to 256 dBm
- Wideband: float: float MS power, measured with the wideband filter (8 MHz) . Range: -256 dBm to 256 dBm
- Out_Of_Tolerance: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (see [CMDLINK: CONFigure:CDMA:MEASi:MEvaluation:SCOut:SPECTrum CMDLINK]) exceeding the specified limits, see 'Limits (Spectrum) '. Range: 0 % to 100 %

class ResultData

Response structure. Fields:

- Reliability: int: decimal
- Narrow_Band: float: float MS power, measured with a filter bandwidth of 1.23 MHz. Range: -256 dBm to 256 dBm
- Wideband: float: float MS power, measured with the wideband filter (8 MHz) . Range: -256 dBm to 256 dBm
- Out_Of_Tolerance: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (see [CMDLINK: CONFigure:CDMA:MEASi:MEvaluation:SCOut:SPECTrum CMDLINK]) exceeding the specified limits, see 'Limits (Spectrum) '. Range: 0 % to 100 %

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:ACP:CURRENT
value: CalculateStruct = driver.multiEval.acp.current.calculate()
```

Returns MS power and the 'out of tolerance' statistical results. For the MS power results, the current, average and maximum values can be retrieved. The 'Out of Tolerance' retrieved via the CURRENT, AVERAGE and MAXimum command are identical. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:ACP:CURRent
value: ResultData = driver.multiEval.acp.current.fetch()
```

Returns MS power and the ‘out of tolerance’ statistical results. For the MS power results, the current, average and maximum values can be retrieved. The ‘Out of Tolerance’ retrieved via the CURRent, AVERAge and MAXimum command are identical. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:ACP:CURRent
value: ResultData = driver.multiEval.acp.current.read()
```

Returns MS power and the ‘out of tolerance’ statistical results. For the MS power results, the current, average and maximum values can be retrieved. The ‘Out of Tolerance’ retrieved via the CURRent, AVERAge and MAXimum command are identical. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.4.2 Average

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:ACP:AVERAge
FETCh:CDMA:MEASurement<Instance>:MEValuation:ACP:AVERAge
CALCulate:CDMA:MEASurement<Instance>:MEValuation:ACP:AVERAge
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal
- Narrow_Band: float: float MS power, measured with a filter bandwidth of 1.23 MHz. Range: -256 dBm to 256 dBm
- Wideband: float: float MS power, measured with the wideband filter (8 MHz) . Range: -256 dBm to 256 dBm
- Out_Of_Tolerance: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (see [CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCOunt:SPECTrum CMDLINK]) exceeding the specified limits, see ‘Limits (Spectrum)’. Range: 0 % to 100 %

class ResultData

Response structure. Fields:

- Reliability: int: decimal

- **Narrow_Band:** float: float MS power, measured with a filter bandwidth of 1.23 MHz. Range: -256 dBm to 256 dBm
- **Wideband:** float: float MS power, measured with the wideband filter (8 MHz) . Range: -256 dBm to 256 dBm
- **Out_Of_Tolerance:** float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (see [CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCOut:SPECTrum CMDLINK]) exceeding the specified limits, see 'Limits (Spectrum) '. Range: 0 % to 100 %

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:ACP:AVERage
value: CalculateStruct = driver.multiEval.acp.average.calculate()
```

Returns MS power and the 'out of tolerance' statistical results. For the MS power results, the current, average and maximum values can be retrieved. The 'Out of Tolerance' retrieved via the CURRent, AVERage and MAXimum command are identical. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:ACP:AVERage
value: ResultData = driver.multiEval.acp.average.fetch()
```

Returns MS power and the 'out of tolerance' statistical results. For the MS power results, the current, average and maximum values can be retrieved. The 'Out of Tolerance' retrieved via the CURRent, AVERage and MAXimum command are identical. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:ACP:AVERage
value: ResultData = driver.multiEval.acp.average.read()
```

Returns MS power and the 'out of tolerance' statistical results. For the MS power results, the current, average and maximum values can be retrieved. The 'Out of Tolerance' retrieved via the CURRent, AVERage and MAXimum command are identical. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.4.3 Maximum

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEValuation:ACP:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEValuation:ACP:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:ACP:MAXimum
```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal
- Narrow_Band: float: float MS power, measured with a filter bandwidth of 1.23 MHz. Range: -256 dBm to 256 dBm
- Wideband: float: float MS power, measured with the wideband filter (8 MHz) . Range: -256 dBm to 256 dBm
- Out_Of_Tolerance: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (see [CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCOut:SPECTrum CMDLINK]) exceeding the specified limits, see 'Limits (Spectrum) '. Range: 0 % to 100 %

class ResultData

Response structure. Fields:

- Reliability: int: decimal
- Narrow_Band: float: float MS power, measured with a filter bandwidth of 1.23 MHz. Range: -256 dBm to 256 dBm
- Wideband: float: float MS power, measured with the wideband filter (8 MHz) . Range: -256 dBm to 256 dBm
- Out_Of_Tolerance: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (see [CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCOut:SPECTrum CMDLINK]) exceeding the specified limits, see 'Limits (Spectrum) '. Range: 0 % to 100 %

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:ACP:MAXimum
value: CalculateStruct = driver.multiEval.acp.maximum.calculate()
```

Returns MS power and the 'out of tolerance' statistical results. For the MS power results, the current, average and maximum values can be retrieved. The 'Out of Tolerance' retrieved via the CURRent, AVERage and MAXimum command are identical. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:ACP:MAXimum
value: ResultData = driver.multiEval.acp.maximum.fetch()
```


Returns MS power and the ‘out of tolerance’ statistical results. For the MS power results, the current, average and maximum values can be retrieved. The ‘Out of Tolerance’ retrieved via the CURRENT, AVERAGE and MAXIMUM command are identical. The values described below are returned by FETCH and READ commands. CALCULATE commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:ACP:MAXimum
value: ResultData = driver.multiEval.acp.maximum.read()
```

Returns MS power and the ‘out of tolerance’ statistical results. For the MS power results, the current, average and maximum values can be retrieved. The ‘Out of Tolerance’ retrieved via the CURRENT, AVERAGE and MAXIMUM command are identical. The values described below are returned by FETCH and READ commands. CALCULATE commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.5 Obw

class Obw

Obw commands group definition. 9 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.obw.clone()
```

Subgroups

7.4.5.1 Current

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:MEvaluation:OBW:CURRENT
FETCH:CDMA:MEASurement<Instance>:MEvaluation:OBW:CURRENT
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:OBW:CURRENT
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEvaluation:SCount:MODulation CMDLINK]) exceeding the specified limit, see ‘Limits (Spectrum)’. Range: 0 % to 100 %, Unit: %

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limit, see 'Limits (Spectrum) '. Range: 0 % to 100 %, Unit: %

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:OBW:CURRENT
value: CalculateStruct = driver.multiEval.obw.current.calculate()
```

Return the current, average and maximum occupied bandwidth value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:OBW:CURRENT
value: ResultData = driver.multiEval.obw.current.fetch()
```

Return the current, average and maximum occupied bandwidth value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:OBW:CURRENT
value: ResultData = driver.multiEval.obw.current.read()
```

Return the current, average and maximum occupied bandwidth value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.5.2 Average**SCPI Commands**

```
READ:CDMA:MEASurement<Instance>:MEValuation:OBW:AVERage
FETCh:CDMA:MEASurement<Instance>:MEValuation:OBW:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:OBW:AVERage
```

class Average

Average commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limit, see 'Limits (Spectrum) '. Range: 0 % to 100 %, Unit: %

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limit, see 'Limits (Spectrum) '. Range: 0 % to 100 %, Unit: %

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:OBW:AVERage
value: CalculateStruct = driver.multiEval.obw.average.calculate()
```

Return the current, average and maximum occupied bandwidth value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:OBW:AVERage
value: ResultData = driver.multiEval.obw.average.fetch()
```

Return the current, average and maximum occupied bandwidth value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEValuation:OBW:AVERage
value: ResultData = driver.multiEval.obw.average.read()
```

Return the current, average and maximum occupied bandwidth value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.5.3 Maximum

SCPI Commands

```

READ:CDMA:MEASurement<Instance>:MEValuation:OBW:MAXimum
FETCh:CDMA:MEASurement<Instance>:MEValuation:OBW:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:OBW:MAXimum

```

class Maximum

Maximum commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limit, see 'Limits (Spectrum) '. Range: 0 % to 100 %, Unit: %

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count ([CMDLINK: CONFigure:CDMA:MEASi:MEValuation:SCount:MODulation CMDLINK]) exceeding the specified limit, see 'Limits (Spectrum) '. Range: 0 % to 100 %, Unit: %

calculate() → CalculateStruct

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:OBW:MAXimum
value: CalculateStruct = driver.multiEval.obw.maximum.calculate()

```

Return the current, average and maximum occupied bandwidth value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:OBW:MAXimum
value: ResultData = driver.multiEval.obw.maximum.fetch()

```

Return the current, average and maximum occupied bandwidth value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:CDMA:MEASurement<Instance>:MEvaluation:OBW:MAXimum
value: ResultData = driver.multiEval.obw.maximum.read()
```

Return the current, average and maximum occupied bandwidth value results. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for ResultData structure arguments.

7.4.6 ListPy

class ListPy

ListPy commands group definition. 455 total commands, 8 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.clone()
```

Subgroups

7.4.6.1 Sreliability

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:SREliability
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SREliability
```

class Sreliability

Sreliability commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SREliability
value: List[enums.ResultStatus2] = driver.multiEval.listPy.sreliability.
    ↪ calculate()
```

Returns the segment reliabilities for all active list mode segments. A common reliability indicator of zero indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments. If you get a non-zero common reliability indicator, you can use this command to retrieve the individual reliability values of all measured segments for further analysis. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return seg_reliability: decimal Comma-separated list of values, one per active segment.
The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error
If a combination of exceptions occurs, the most severe error is indicated.

fetch() → List[int]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SRELiability
value: List[int] = driver.multiEval.listPy.sreliability.fetch()
```

Returns the segment reliabilities for all active list mode segments. A common reliability indicator of zero indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments. If you get a non-zero common reliability indicator, you can use this command to retrieve the individual reliability values of all measured segments for further analysis. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return seg_reliability: decimal Comma-separated list of values, one per active segment.
 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error
 If a combination of exceptions occurs, the most severe error is indicated.

7.4.6.2 Modulation

class Modulation

Modulation commands group definition. 124 total commands, 17 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.clone()
```

Subgroups

7.4.6.2.1 Otolerance

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:OTOLerance
```

class Otolerance

Otolerance commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[int]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:OTOLerance
value: List[int] = driver.multiEval.listPy.modulation.otolerance.fetch()
```

Returns the out of tolerance percentages in the OBW measurement for all active list mode segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return out_of_tol_count: decimal The percentage of measurement intervals of the statistic count exceeding the specified limits. Comma-separated list of values, one per active segment. Range: 0 % to 100 % , Unit: %

7.4.6.2.2 StCount

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:STCount
```

class StCount

StCount commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[int]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:STCount
value: List[int] = driver.multiEval.listPy.modulation.stCount.fetch()
```

Returns the statistic count in the modulation measurement for all active list mode segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return statistic_count: decimal The number of evaluated valid slots. Comma-separated list of values, one per active segment. Range: 0 to 1000

7.4.6.2.3 Evm

class Evm

Evm commands group definition. 16 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.evm.clone()
```

Subgroups

7.4.6.2.3.1 Rms

class Rms

Rms commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.evm.rms.clone()
```

Subgroups

7.4.6.2.3.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:CURRENT
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:CURRENT
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:CURRENT
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.current.
↪calculate()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:CURRENT
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.current.fetch()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.3.3 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:AVERage
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.average.
↪calculate()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:AVERage
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.average.fetch()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.3.4 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.maximum.
↪calculate()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.maximum.fetch()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.3.5 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.standardDev.
↪calculate()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.standardDev.
↪fetch()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.3.6 Peak

class Peak

Peak commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.evm.peak.clone()
```

Subgroups

7.4.6.2.3.7 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:EVM:PEAK:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:EVM:PEAK:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:MODulation:EVM:PEAK:CURRent
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.current.
↪calculate()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:MODulation:EVM:PEAK:CURRent
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.current.fetch()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.3.8 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:PEAK:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:PEAK:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:AVERage
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.average.
↪calculate()

```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:AVERage
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.average.fetch()

```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.3.9 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:EVM:PEAK:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:EVM:PEAK:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:MODulation:EVM:PEAK:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.maximum.
↪calculate()

```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:MODulation:EVM:PEAK:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.maximum.fetch()

```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.3.10 StandardDev

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:EVM:PEAK:SDEVIation
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:EVM:PEAK:SDEVIation

```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.standardDev.
↪calculate()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.standardDev.
↪fetch()
```

Returns EVM peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.4 Merror

class Merror

Merror commands group definition. 16 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.merror.clone()
```

Subgroups

7.4.6.2.4.1 Rms

class Rms

Rms commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.merror.rms.clone()
```

Subgroups

7.4.6.2.4.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:CURRent
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.current.
↪calculate()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:CURRent
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.current.
↪fetch()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.4.3 Average

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:MERRor:RMS:AVERage
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.average.
↳calculate()

```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:MERRor:RMS:AVERage
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.average.
↳fetch()

```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.4.4 Maximum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]


```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.maximum.
↪calculate()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.maximum.
↪fetch()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.4.5 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:SDEVIation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:SDEVIation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:SDEVIation
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.standardDev.
↪calculate()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.standardDev.
↪fetch()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_rms: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.4.6 Peak

class Peak

Peak commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.merror.peak.clone()
```

Subgroups

7.4.6.2.4.7 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:PEAK:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:PEAK:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:PEAK:CURRent
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.current.
↪calculate()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:PEAK:CURRent
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.current.
↪fetch()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.4.8 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:PEAK:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:PEAK:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:PEAK:AVERage
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.average.
↪calculate()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:PEAK:AVERage
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.average.
↪fetch()
```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.4.9 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:PEAK:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:PEAK:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:PEAK:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.maximum.
↪calculate()

```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:MERRor:PEAK:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.maximum.
↪fetch()

```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.4.10 StandardDev

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:MERRor:PEAK:SDEViation
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:MERRor:PEAK:SDEViation

```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:LIST:MODulation:MERRor:PEAK:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.standardDev.
↳calculate()

```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEValuation:LIST:MODulation:MERRor:PEAK:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.standardDev.
↳fetch()

```

Returns peak and RMS magnitude error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return merr_peak: float Comma-separated list of values, one per active segment. Range:
0 % to 100 %, Unit: %

7.4.6.2.5 Perror

class Perror

Perror commands group definition. 16 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.perror.clone()
```

Subgroups

7.4.6.2.5.1 Rms

class Rms

Rms commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.perror.rms.clone()
```

Subgroups

7.4.6.2.5.2 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:PERRor:RMS:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:PERRor:RMS:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:MODulation:PERRor:RMS:CURRent
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.current.
↪calculate()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_rms: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:CURRent
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.current.
↪fetch()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_rms: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

7.4.6.2.5.3 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:AVERage
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.average.
↪calculate()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_rms: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:AVERage
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.average.
↪fetch()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_rms: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

7.4.6.2.5.4 Maximum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.maximum.
↪calculate()

```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_rms: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.maximum.
↪fetch()

```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_rms: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

7.4.6.2.5.5 StandardDev

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:SDEviation

```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]


```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:SDEVIation
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.standardDev.
↪calculate()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_rms: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:SDEVIation
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.standardDev.
↪fetch()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_rms: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

7.4.6.2.5.6 Peak

class Peak

Peak commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.perror.peak.clone()
```

Subgroups

7.4.6.2.5.7 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:PERRor:PEAK:CURRent
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.current.
↳calculate()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_peak: float Comma-separated list of values, one per active segment. Range: 0 deg to 180 deg , Unit: deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:PERRor:PEAK:CURRent
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.current.
↳fetch()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_peak: float Comma-separated list of values, one per active segment. Range: 0 deg to 180 deg , Unit: deg

7.4.6.2.5.8 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:PERRor:PEAK:AVERage
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.average.
↳calculate()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_peak: float Comma-separated list of values, one per active segment. Range: 0 deg to 180 deg , Unit: deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:PEAK:AVErage
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.average.
↪fetch()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_peak: float Comma-separated list of values, one per active segment. Range: 0 deg to 180 deg , Unit: deg

7.4.6.2.5.9 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: enums.ResultStatus2: decimal See ‘Reliability Indicator’.
- Perr_Peak: List[float]: float Comma-separated list of values, one per active segment. Range: 0 deg to 180 deg , Unit: deg

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:PEAK:MAXimum
value: CalculateStruct = driver.multiEval.listPy.modulation.perror.peak.maximum.
↪calculate()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:PEAK:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.maximum.
↪fetch()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_peak: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

7.4.6.2.5.10 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:PEAK:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.standardDev.
↪calculate()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_peak: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PERRor:PEAK:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.standardDev.
↪fetch()
```

Returns peak and RMS phase error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return perr_peak: float Comma-separated list of values, one per active segment. Range:
0 deg to 180 deg , Unit: deg

7.4.6.2.6 IqOffset

class IqOffset

IqOffset commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.iqOffset.clone()
```

Subgroups

7.4.6.2.6.1 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:IQOFfset:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:IQOFfset:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:MODulation:IQOFfset:CURRent
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.current.
↪calculate()
```

Returns IQ origin offset (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per active segment. Range:
-100 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:MODulation:IQOFfset:CURRent
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.current.fetch()
```

Returns IQ origin offset (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per active segment. Range:
-100 dB to 0 dB , Unit: dB

7.4.6.2.6.2 Average

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQOffset:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQOffset:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQOffset:AVERage
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.average.
↪calculate()

```

Returns IQ origin offset (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per active segment. Range: -100 dB to 0 dB , Unit: dB

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQOffset:AVERage
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.average.fetch()

```

Returns IQ origin offset (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per active segment. Range: -100 dB to 0 dB , Unit: dB

7.4.6.2.6.3 Maximum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQOffset:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQOffset:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:IQOffset:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.maximum.
↳calculate()
```

Returns IQ origin offset (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per active segment. Range: -100 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:IQOffset:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.maximum.fetch()
```

Returns IQ origin offset (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per active segment. Range: -100 dB to 0 dB , Unit: dB

7.4.6.2.6.4 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQOffset:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQOffset:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:IQOffset:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.standardDev.
↳calculate()
```

Returns IQ origin offset (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per active segment. Range: -100 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQOffset:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.standardDev.
↪fetch()
```

Returns IQ origin offset (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per active segment. Range: -100 dB to 0 dB , Unit: dB

7.4.6.2.7 IqImbalance

class IqImbalance

IqImbalance commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.iqImbalance.clone()
```

Subgroups

7.4.6.2.7.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:CURRent
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.current.
↪calculate()
```

Returns IQ imbalance (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per active segment.
Range: -100 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:CURRENT
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.current.
↪fetch()
```

Returns IQ imbalance (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per active segment.
Range: -100 dB to 0 dB , Unit: dB

7.4.6.2.7.2 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:AVERage
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.average.
↪calculate()
```

Returns IQ imbalance (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per active segment.
Range: -100 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:AVERage
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.average.
↪fetch()
```

Returns IQ imbalance (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per active segment.
Range: -100 dB to 0 dB , Unit: dB

7.4.6.2.7.3 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.maximum.
↪calculate()

```

Returns IQ imbalance (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per active segment.
Range: -100 dB to 0 dB , Unit: dB

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.maximum.
↪fetch()

```

Returns IQ imbalance (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per active segment.
Range: -100 dB to 0 dB , Unit: dB

7.4.6.2.7.4 StandardDev

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:SDEviation

```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.standardDev.
↪calculate()

```

Returns IQ imbalance (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per active segment.
Range: -100 dB to 0 dB , Unit: dB

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.standardDev.
↪fetch()

```

Returns IQ imbalance (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per active segment.
Range: -100 dB to 0 dB , Unit: dB

7.4.6.2.8 FreqError

class FreqError

FreqError commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.freqError.clone()
```

Subgroups

7.4.6.2.8.1 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:FERRor:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:FERRor:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:MODulation:FERRor:CURRent
value: List[float] = driver.multiEval.listPy.modulation.freqError.current.
↪calculate()
```

Returns carrier frequency error peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cfreq_error: float Comma-separated list of values, one per active segment.
Range: -5000 Hz to 5000 Hz , Unit: Hz

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:MODulation:FERRor:CURRent
value: List[float] = driver.multiEval.listPy.modulation.freqError.current.
↪fetch()
```

Returns carrier frequency error peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cfreq_error: float Comma-separated list of values, one per active segment.
Range: -5000 Hz to 5000 Hz , Unit: Hz

7.4.6.2.8.2 Average

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:FERRor:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:FERRor:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:LIST:MODulation:FERRor:AVERage
value: List[float] = driver.multiEval.listPy.modulation.freqError.average.
↳calculate()

```

Returns carrier frequency error peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cfreq_error: float Comma-separated list of values, one per active segment.
Range: -5000 Hz to 5000 Hz , Unit: Hz

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>
↳:MEValuation:LIST:MODulation:FERRor:AVERage
value: List[float] = driver.multiEval.listPy.modulation.freqError.average.
↳fetch()

```

Returns carrier frequency error peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cfreq_error: float Comma-separated list of values, one per active segment.
Range: -5000 Hz to 5000 Hz , Unit: Hz

7.4.6.2.8.3 Maximum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:FERRor:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:FERRor:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:FERRor:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.freqError.maximum.
↪calculate()
```

Returns carrier frequency error peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cfreq_error: float Comma-separated list of values, one per active segment.
Range: -5000 Hz to 5000 Hz , Unit: Hz

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:FERRor:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.freqError.maximum.
↪fetch()
```

Returns carrier frequency error peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cfreq_error: float Comma-separated list of values, one per active segment.
Range: -5000 Hz to 5000 Hz , Unit: Hz

7.4.6.2.8.4 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:FERRor:SDEViation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:FERRor:SDEViation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:FERRor:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.freqError.standardDev.
↪calculate()
```

Returns carrier frequency error peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cfreq_error: float Comma-separated list of values, one per active segment.
Range: -5000 Hz to 5000 Hz , Unit: Hz

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:FERRor:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.freqError.standardDev.
↪fetch()
```

Returns carrier frequency error peak and RMS (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return cfreq_error: float Comma-separated list of values, one per active segment.
Range: -5000 Hz to 5000 Hz , Unit: Hz

7.4.6.2.9 Terror

class Terror

Terror commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.terror.clone()
```

Subgroups

7.4.6.2.9.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TERRor:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TERRor:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:TERRor:CURRent
value: List[float] = driver.multiEval.listPy.modulation.terror.current.
↪calculate()
```

Returns transmit time error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return trans_time_error: float Comma-separated list of values, one per active segment.
Range: -100 µs to 100 µs, Unit: µs

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:TERRor:CURRent
value: List[float] = driver.multiEval.listPy.modulation.terror.current.fetch()
```

Returns transmit time error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return trans_time_error: float Comma-separated list of values, one per active segment.
Range: -100 µs to 100 µs, Unit: µs

7.4.6.2.9.2 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TERRor:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TERRor:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:TERRor:AVERage
value: List[float] = driver.multiEval.listPy.modulation.terror.average.
↪calculate()
```

Returns transmit time error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return trans_time_error: float Comma-separated list of values, one per active segment.
Range: -100 µs to 100 µs, Unit: µs

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:TERRor:AVERage
value: List[float] = driver.multiEval.listPy.modulation.terror.average.fetch()
```

Returns transmit time error (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return trans_time_error: float Comma-separated list of values, one per active segment.
Range: -100 μ s to 100 μ s, Unit: μ s

7.4.6.2.9.3 Maximum

SCPI Commands

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TERRor:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TERRor:MAXimum

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:TERRor:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.terror.maximum.
↪calculate()
```

Returns transmit time error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return trans_time_error: float Comma-separated list of values, one per active segment.
Range: -100 μ s to 100 μ s, Unit: μ s

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:TERRor:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.terror.maximum.fetch()
```

Returns transmit time error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return trans_time_error: float Comma-separated list of values, one per active segment.
Range: -100 μ s to 100 μ s, Unit: μ s

7.4.6.2.9.4 StandardDev

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TERRor:SDEViation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TERRor:SDEViation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:TERRor:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.terror.standardDev.
↳calculate()
```

Returns transmit time error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return trans_time_error: float Comma-separated list of values, one per active segment.
Range: -100 µs to 100 µs, Unit: µs

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:TERRor:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.terror.standardDev.
↳fetch()
```

Returns transmit time error (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return trans_time_error: float Comma-separated list of values, one per active segment.
Range: -100 µs to 100 µs, Unit: µs

7.4.6.2.10 Wquality

class Wquality

Wquality commands group definition. 12 total commands, 6 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.wquality.clone()
```

Subgroups

7.4.6.2.10.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:WQuality:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:WQuality:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:WQuality:CURRent
value: List[float] = driver.multiEval.listPy.modulation.wquality.current.
↪calculate()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_quality: float Comma-separated list of values, one per active segment.
Range: -100 to 100

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:WQuality:CURRent
value: List[float] = driver.multiEval.listPy.modulation.wquality.current.fetch()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_quality: float Comma-separated list of values, one per active segment.
Range: -100 to 100

7.4.6.2.10.2 Pmax

class Pmax

Pmax commands group definition. 2 total commands, 1 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.wquality.pmax.clone()
```

Subgroups

7.4.6.2.10.3 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:WQuality:PMAX:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:WQuality:PMAX:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:WQuality:PMAX:CURRent
value: List[float] = driver.multiEval.listPy.modulation.wquality.pmax.current.
↳calculate()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_qual_max_pow: float Comma-separated list of values, one per active segment. Range: -100 to 100

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:WQuality:PMAX:CURRent
value: List[float] = driver.multiEval.listPy.modulation.wquality.pmax.current.
↳fetch()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_qual_max_pow: float Comma-separated list of values, one per active segment. Range: -100 to 100

7.4.6.2.10.4 Pmin

class Pmin

Pmin commands group definition. 2 total commands, 1 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.wquality.pmin.clone()
```

Subgroups

7.4.6.2.10.5 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:WQuality:PMIN:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:WQuality:PMIN:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:WQuality:PMIN:CURRent
value: List[float] = driver.multiEval.listPy.modulation.wquality.pmin.current.
↳calculate()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_qual_min_power: float Comma-separated list of values, one per active segment. Range: -100 to 100

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:WQuality:PMIN:CURRent
value: List[float] = driver.multiEval.listPy.modulation.wquality.pmin.current.
↳fetch()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_qual_min_power: float Comma-separated list of values, one per active segment. Range: -100 to 100

7.4.6.2.10.6 Average

SCPI Commands

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:WQQuality:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:WQQuality:AVERage

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:WQQuality:AVERage
value: List[float] = driver.multiEval.listPy.modulation.wquality.average.
↳calculate()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_quality: float Comma-separated list of values, one per active segment.
Range: -100 to 100

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:WQQuality:AVERage
value: List[float] = driver.multiEval.listPy.modulation.wquality.average.fetch()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_quality: float Comma-separated list of values, one per active segment.
Range: -100 to 100

7.4.6.2.10.7 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:WQQuality:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:WQQuality:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:LIST:MODulation:WQQuality:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.wquality.maximum.
↳calculate()

```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_quality: float Comma-separated list of values, one per active segment.
Range: -100 to 100

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEValuation:LIST:MODulation:WQQuality:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.wquality.maximum.fetch()

```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_quality: float Comma-separated list of values, one per active segment.
Range: -100 to 100

7.4.6.2.10.8 StandardDev

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:WQQuality:SDEVIation
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:WQQuality:SDEVIation

```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:WQQuality:SDEVIation
value: List[float] = driver.multiEval.listPy.modulation.wquality.standardDev.
↳calculate()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_quality: float Comma-separated list of values, one per active segment.
Range: -100 to 100

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:WQQuality:SDEVIation
value: List[float] = driver.multiEval.listPy.modulation.wquality.standardDev.
↳fetch()
```

Returns waveform quality (statistical) values for all active list mode segments - overall, at minimum and at maximum MS power. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wav_quality: float Comma-separated list of values, one per active segment.
Range: -100 to 100

7.4.6.2.11 PwBand

class PwBand

PwBand commands group definition. 10 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.pwBand.clone()
```

Subgroups

7.4.6.2.11.1 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PWBand:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PWBand:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PWBand:CURRENT
value: List[float] = driver.multiEval.listPy.modulation.pwBand.current.
↪calculate()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PWBand:CURRENT
value: List[float] = driver.multiEval.listPy.modulation.pwBand.current.fetch()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.11.2 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PWBand:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PWBand:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PWBand:AVERage
value: List[float] = driver.multiEval.listPy.modulation.pwBand.average.
↪calculate()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned

by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PWBand:AVERage
value: List[float] = driver.multiEval.listPy.modulation.pwBand.average.fetch()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.11.3 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PWBand:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PWBand:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PWBand:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.pwBand.maximum.
↪calculate()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PWBand:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.pwBand.maximum.fetch()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.11.4 Minimum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PWBand:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PWBand:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PWBand:MINimum
value: List[float] = driver.multiEval.listPy.modulation.pwBand.minimum.
↪calculate()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PWBand:MINimum
value: List[float] = driver.multiEval.listPy.modulation.pwBand.minimum.fetch()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.11.5 StandardDev

SCPI Commands

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:PWBand:SDEViation CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:PWBand:SDEViation
--

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

<pre># SCPI: CALCulate:CDMA:MEASurement<Instance> ↪:MEValuation:LIST:MODulation:PWBand:SDEViation value: List[float] = driver.multiEval.listPy.modulation.pwBand.standardDev. ↪calculate()</pre>
--

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

<pre># SCPI: FETCh:CDMA:MEASurement<Instance> ↪:MEValuation:LIST:MODulation:PWBand:SDEViation value: List[float] = driver.multiEval.listPy.modulation.pwBand.standardDev. ↪fetch()</pre>
--

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_wideband: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.12 PnBand

class PnBand

PnBand commands group definition. 10 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.modulation.pnBand.clone()
```

Subgroups

7.4.6.2.12.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PNBand:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PNBand:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:PNBand:CURRent
value: List[float] = driver.multiEval.listPy.modulation.pnBand.current.
↳calculate()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:MODulation:PNBand:CURRent
value: List[float] = driver.multiEval.listPy.modulation.pnBand.current.fetch()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.12.2 Average

SCPI Commands

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:PNBand:AVERage CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:PNBand:AVERage
--

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

<pre># SCPI: CALCulate:CDMA:MEASurement<Instance> ↪:MEValuation:LIST:MODulation:PNBand:AVERage value: List[float] = driver.multiEval.listPy.modulation.pnBand.average. ↪calculate()</pre>

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

<pre># SCPI: FETCh:CDMA:MEASurement<Instance> ↪:MEValuation:LIST:MODulation:PNBand:AVERage value: List[float] = driver.multiEval.listPy.modulation.pnBand.average.fetch()</pre>

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.12.3 Maximum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PNBand:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PNBand:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PNBand:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.pnBand.maximum.
↪calculate()

```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PNBand:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.pnBand.maximum.fetch()

```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.12.4 Minimum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PNBand:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PNBand:MINimum

```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PNBand:MINimum
value: List[float] = driver.multiEval.listPy.modulation.pnBand.minimum.
↪calculate()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PNBand:MINimum
value: List[float] = driver.multiEval.listPy.modulation.pnBand.minimum.fetch()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.12.5 StandardDev

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PNBand:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PNBand:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PNBand:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.pnBand.standardDev.
↪calculate()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned

by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:MODulation:PNBand:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.pnBand.standardDev.
↪fetch()
```

Returns MS narrowband and wideband power (statistical) values for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return ms_power_1_m_23: float Comma-separated list of values, one per active segment. Range: -100 dBm to 50 dBm , Unit: dBm

7.4.6.2.13 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg

- **Perr_Peak:** List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEVIation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset:** List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- **Iq_Imbalance:** List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- **Cfreq_Error:** List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- **Trans_Time_Err:** List[float]: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- **Ms_Power_1_M_23:** List[float]: No parameter help available
- **Ms_Power_Wideband:** List[float]: No parameter help available
- **Wav_Quality:** List[float]: float Waveform quality Range: 0 to 1
- **Wav_Qual_Max_Pow:** List[float]: No parameter help available
- **Wav_Qual_Min_Power:** List[float]: No parameter help available
- **Out_Of_Tol_Count:** List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- **Reliability:** int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability:** List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Evm_Rms:** List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- **Evm_Peak:** List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- **Merr_Rms:** List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- **Merr_Peak:** List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEVIation: 0 % to 50 %) , Unit: %
- **Perr_Rms:** List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- **Perr_Peak:** List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEVIation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset:** List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- **Iq_Imbalance:** List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- **Cfreq_Error:** List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- **Trans_Time_Err:** List[float]: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- **Ms_Power_1_M_23:** List[float]: No parameter help available
- **Ms_Power_Wideband:** List[float]: No parameter help available

- Wav_Quality: List[float]: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: List[float]: No parameter help available
- Wav_Qual_Min_Power: List[float]: No parameter help available
- Out_Of_Tol_Count: List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:CURRENT
value: CalculateStruct = driver.multiEval.listPy.modulation.current.calculate()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:CURRENT
value: FetchStruct = driver.multiEval.listPy.modulation.current.fetch()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.2.14 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: List[float]: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: List[float]: No parameter help available
- Ms_Power_Wideband: List[float]: No parameter help available
- Wav_Quality: List[float]: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: List[float]: No parameter help available
- Wav_Qual_Min_Power: List[float]: No parameter help available
- Out_Of_Tol_Count: List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: List[float]: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: List[float]: No parameter help available
- Ms_Power_Wideband: List[float]: No parameter help available
- Wav_Quality: List[float]: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: List[float]: No parameter help available
- Wav_Qual_Min_Power: List[float]: No parameter help available
- Out_Of_Tol_Count: List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:AVERage
value: CalculateStruct = driver.multiEval.listPy.modulation.average.calculate()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ...,

{...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:AVERage
value: FetchStruct = driver.multiEval.listPy.modulation.average.fetch()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.2.15 Maximum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERAGE: 0% to 100 %, SDEviation: 0 % to 50 %) , Unit: %
- Perr_Rms: List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg

- **Perr_Peak:** List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEVIation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset:** List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- **Iq_Imbalance:** List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- **Cfreq_Error:** List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- **Trans_Time_Err:** List[float]: float Transmit time error Range: -100 μ s to 100 μ s, Unit: μ s
- **Ms_Power_1_M_23:** List[float]: No parameter help available
- **Ms_Power_Wideband:** List[float]: No parameter help available
- **Wav_Quality:** List[float]: float Waveform quality Range: 0 to 1
- **Wav_Qual_Max_Pow:** List[float]: No parameter help available
- **Wav_Qual_Min_Power:** List[float]: No parameter help available
- **Out_Of_Tol_Count:** List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- **Reliability:** int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability:** List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Evm_Rms:** List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- **Evm_Peak:** List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- **Merr_Rms:** List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- **Merr_Peak:** List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEVIation: 0 % to 50 %) , Unit: %
- **Perr_Rms:** List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- **Perr_Peak:** List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEVIation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset:** List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- **Iq_Imbalance:** List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- **Cfreq_Error:** List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- **Trans_Time_Err:** List[float]: float Transmit time error Range: -100 μ s to 100 μ s, Unit: μ s
- **Ms_Power_1_M_23:** List[float]: No parameter help available
- **Ms_Power_Wideband:** List[float]: No parameter help available

- Wav_Quality: List[float]: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: List[float]: No parameter help available
- Wav_Qual_Min_Power: List[float]: No parameter help available
- Out_Of_Tol_Count: List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOunt: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:MAXimum
value: CalculateStruct = driver.multiEval.listPy.modulation.maximum.calculate()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:MAXimum
value: FetchStruct = driver.multiEval.listPy.modulation.maximum.fetch()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.2.16 Minimum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:MINimum

```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: List[float]: float Transmit time error Range: -100 μs to 100 μs, Unit: μs
- Ms_Power_1_M_23: List[float]: No parameter help available
- Ms_Power_Wideband: List[float]: No parameter help available
- Wav_Quality: List[float]: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: List[float]: No parameter help available
- Wav_Qual_Min_Power: List[float]: No parameter help available
- Out_Of_Tol_Count: List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: List[float]: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: List[float]: No parameter help available
- Ms_Power_Wideband: List[float]: No parameter help available
- Wav_Quality: List[float]: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: List[float]: No parameter help available
- Wav_Qual_Min_Power: List[float]: No parameter help available
- Out_Of_Tol_Count: List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:MINimum
value: CalculateStruct = driver.multiEval.listPy.modulation.minimum.calculate()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ...,

{...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MINimum
value: FetchStruct = driver.multiEval.listPy.modulation.minimum.fetch()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.2.17 StandardDev

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERAGE: 0% to 100 %, SDEviation: 0 % to 50 %) , Unit: %
- Perr_Rms: List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg

- **Perr_Peak:** List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEVIation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset:** List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- **Iq_Imbalance:** List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- **Cfreq_Error:** List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- **Trans_Time_Err:** List[float]: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- **Ms_Power_1_M_23:** List[float]: No parameter help available
- **Ms_Power_Wideband:** List[float]: No parameter help available
- **Wav_Quality:** List[float]: float Waveform quality Range: 0 to 1
- **Wav_Qual_Max_Pow:** List[float]: No parameter help available
- **Wav_Qual_Min_Power:** List[float]: No parameter help available
- **Out_Of_Tol_Count:** List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- **Reliability:** int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability:** List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Evm_Rms:** List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- **Evm_Peak:** List[float]: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- **Merr_Rms:** List[float]: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- **Merr_Peak:** List[float]: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEVIation: 0 % to 50 %) , Unit: %
- **Perr_Rms:** List[float]: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- **Perr_Peak:** List[float]: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEVIation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset:** List[float]: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- **Iq_Imbalance:** List[float]: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- **Cfreq_Error:** List[float]: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- **Trans_Time_Err:** List[float]: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- **Ms_Power_1_M_23:** List[float]: No parameter help available
- **Ms_Power_Wideband:** List[float]: No parameter help available

- Wav_Quality: List[float]: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: List[float]: No parameter help available
- Wav_Qual_Min_Power: List[float]: No parameter help available
- Out_Of_Tol_Count: List[float]: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:LIST:MODulation:SDEviation
value: CalculateStruct = driver.multiEval.listPy.modulation.standardDev.
↳calculate()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:SDEviation
value: FetchStruct = driver.multiEval.listPy.modulation.standardDev.fetch()
```

Returns modulation single value results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3 Segment<Segment>

RepCap Settings

```
# Range: Nr1 .. Nr200
rc = driver.multiEval.listPy.segment.repcap_segment_get()
driver.multiEval.listPy.segment.repcap_segment_set(repcap.Segment.Nr1)
```

class Segment

Segment commands group definition. 61 total commands, 6 Sub-groups, 0 group commands Repeated Capability: Segment, default value after init: Segment.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.segment.clone()
```

Subgroups

7.4.6.3.1 Modulation

class Modulation

Modulation commands group definition. 10 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.segment.modulation.clone()
```

Subgroups

7.4.6.3.1.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:MODulation:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:MODulation:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.

- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Power: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNt: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg

- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Power: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:MODulation:CURRent
value: CalculateStruct = driver.multiEval.listPy.segment.modulation.current.
↳calculate(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEViation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:MODulation:CURRent
value: FetchStruct = driver.multiEval.listPy.segment.modulation.current.
↳fetch(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEViation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by

two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.1.2 Average

SCPI Commands

FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:MODulation:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:MODulation:AVERage

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Power: float: No parameter help available

- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Power: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:MODulation:AVERage
value: CalculateStruct = driver.multiEval.listPy.segment.modulation.average.
↳calculate(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:MODulation:AVERage
value: FetchStruct = driver.multiEval.listPy.segment.modulation.average.
↳fetch(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.1.3 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:MODulation:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:MODulation:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.

- Evm_Rms: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Power: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz

- **Trans_Time_Err:** float: float Transmit time error Range: -100 μ s to 100 μ s, Unit: μ s
- **Ms_Power_1_M_23:** float: No parameter help available
- **Ms_Power_Wideband:** float: No parameter help available
- **Wav_Quality:** float: float Waveform quality Range: 0 to 1
- **Wav_Qual_Max_Pow:** float: No parameter help available
- **Wav_Qual_Min_Power:** float: No parameter help available
- **Out_Of_Tol_Count:** float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) \rightarrow CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:MODulation:MAXimum
value: CalculateStruct = driver.multiEval.listPy.segment.modulation.maximum.
↪calculate(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEViation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) \rightarrow FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:MODulation:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.modulation.maximum.
↪fetch(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEViation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.1.4 Minimum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:MODulation:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:MODulation:MINimum

```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %), Unit: %
- Perr_Rms: float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 μs to 100 μs, Unit: μs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Pow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- Perr_Peak: float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- Iq_Offset: float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- Iq_Imbalance: float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- Cfreq_Error: float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- Trans_Time_Err: float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- Ms_Power_1_M_23: float: No parameter help available
- Ms_Power_Wideband: float: No parameter help available
- Wav_Quality: float: float Waveform quality Range: 0 to 1
- Wav_Qual_Max_Pow: float: No parameter help available
- Wav_Qual_Min_Pow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:MODulation:MINimum
value: CalculateStruct = driver.multiEval.listPy.segment.modulation.minimum.
↪calculate(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum and SDEViation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:MODulation:MINimum
value: FetchStruct = driver.multiEval.listPy.segment.modulation.minimum.
↪fetch(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum and SDEViation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.1.5 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:MODulation:SDEViation
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>
↪:MODulation:SDEViation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Evm_Rms: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- Merr_Rms: float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- Merr_Peak: float: float Magnitude error peak value. Range: -100 % to +100 % (AVERAGE: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Perr_Rms: float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg

- **Perr_Peak:** float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset:** float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- **Iq_Imbalance:** float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- **Cfreq_Error:** float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- **Trans_Time_Err:** float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- **Ms_Power_1_M_23:** float: No parameter help available
- **Ms_Power_Wideband:** float: No parameter help available
- **Wav_Quality:** float: float Waveform quality Range: 0 to 1
- **Wav_Qual_Max_Pow:** float: No parameter help available
- **Wav_Qual_Min_Pow:** float: No parameter help available
- **Out_Of_Tol_Count:** float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- **Reliability:** int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability:** int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Evm_Rms:** float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- **Evm_Peak:** float: float Error vector magnitude RMS and peak value. Range: 0 % to 100 %, Unit: %
- **Merr_Rms:** float: float Magnitude error RMS value. Range: 0 % to 100 %, Unit: %
- **Merr_Peak:** float: float Magnitude error peak value. Range: -100 % to +100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- **Perr_Rms:** float: float Phase error RMS value. Range: 0 deg to 180 deg , Unit: deg
- **Perr_Peak:** float: float Phase error peak value. Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg
- **Iq_Offset:** float: float I/Q origin offset Range: -100 dB to 0 dB , Unit: dB
- **Iq_Imbalance:** float: float I/Q imbalance Range: -100 dB to 0 dB , Unit: dB
- **Cfreq_Error:** float: float Carrier frequency error Range: -5000 Hz to 5000 Hz , Unit: Hz
- **Trans_Time_Err:** float: float Transmit time error Range: -100 µs to 100 µs, Unit: µs
- **Ms_Power_1_M_23:** float: No parameter help available
- **Ms_Power_Wideband:** float: No parameter help available
- **Wav_Quality:** float: float Waveform quality Range: 0 to 1
- **Wav_Qual_Max_Pow:** float: No parameter help available

- Wav_Qual_Min_Pow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:MODulation:SDEviation
value: CalculateStruct = driver.multiEval.listPy.segment.modulation.standardDev.
↳calculate(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:MODulation:SDEviation
value: FetchStruct = driver.multiEval.listPy.segment.modulation.standardDev.
↳fetch(segment = repcap.Segment.Default)
```

Returns modulation single value results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2 Acp

class Acp

Acp commands group definition. 20 total commands, 6 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.segment.acp.clone()
```

Subgroups

7.4.6.3.2.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Current_Acp: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFIGure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.

- **Seg_Reliability**: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- **Current_Acp**: List[float]: No parameter help available
- **Ms_Power_Wide**: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Ms_Power_Narrow**: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm
- **Out_Of_Tol_Count**: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFIGure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count**: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:CURRENT
value: CalculateStruct = driver.multiEval.listPy.segment.acp.current.
↳calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERAGE, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:CURRENT
value: FetchStruct = driver.multiEval.listPy.segment.acp.current.fetch(segment_
↳= repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERAGE, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2.2 Extended

class Extended

Extended commands group definition. 10 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.segment.acp.extended.clone()
```

Subgroups

7.4.6.3.2.3 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:EXTended:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>
↪:ACP:EXTended:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Current_Acp: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: float: No parameter help available
- Ms_Power_Narrow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFIGure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.

- **Seg_Reliability**: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- **Current_Acp**: List[float]: No parameter help available
- **Ms_Power_Wide**: float: No parameter help available
- **Ms_Power_Narrow**: float: No parameter help available
- **Out_Of_Tol_Count**: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFIGure:CDMA:MEASure:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count**: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:EXTended:CURRent
value: CalculateStruct = driver.multiEval.listPy.segment.acp.extended.current.
↳calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERAGE, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:EXTended:CURRent
value: FetchStruct = driver.multiEval.listPy.segment.acp.extended.current.
↳fetch(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERAGE, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2.4 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:EXTended:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>
↳:ACP:EXTended:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Average_Acp: List[float]: No parameter help available
- Ms_Power_Wide: float: No parameter help available
- Ms_Power_Narrow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Average_Acp: List[float]: No parameter help available
- Ms_Power_Wide: float: No parameter help available
- Ms_Power_Narrow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:EXTended:AVERage
value: CalculateStruct = driver.multiEval.listPy.segment.acp.extended.average.
↳calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:EXTended:AVERage
value: FetchStruct = driver.multiEval.listPy.segment.acp.extended.average.
↳fetch(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2.5 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:EXTended:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>
↳:ACP:EXTended:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Maximum_Acp: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: float: No parameter help available
- Ms_Power_Narrow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Maximum_Acp: List[float]: No parameter help available
- Ms_Power_Wide: float: No parameter help available
- Ms_Power_Narrow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:ACP:EXTended:MAXimum
value: CalculateStruct = driver.multiEval.listPy.segment.acp.extended.maximum.
↪calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERAGE, MINimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:ACP:EXTended:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.acp.extended.maximum.
↪fetch(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2.6 Minimum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:EXTended:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>
↪:ACP:EXTended:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Minimum_Acp: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: float: No parameter help available
- Ms_Power_Narrow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Minimum_Acp: List[float]: No parameter help available
- Ms_Power_Wide: float: No parameter help available
- Ms_Power_Narrow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:EXTended:MINimum
value: CalculateStruct = driver.multiEval.listPy.segment.acp.extended.minimum.
↳calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:EXTended:MINimum
value: FetchStruct = driver.multiEval.listPy.segment.acp.extended.minimum.
↳fetch(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for

each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2.7 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>
↪:ACP:EXTended:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>
↪:ACP:EXTended:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Std_Dev_Acp: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: float: No parameter help available
- Ms_Power_Narrow: float: No parameter help available
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEvaluation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Std_Dev_Acp: List[float]: No parameter help available
- Ms_Power_Wide: float: No parameter help available
- Ms_Power_Narrow: float: No parameter help available

- **Out_Of_Tol_Count**: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count**: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGMENT<nr>
↪:ACP:EXTended:SDEviation
value: CalculateStruct = driver.multiEval.listPy.segment.acp.extended.
↪standardDev.calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGMENT<nr>
↪:ACP:EXTended:SDEviation
value: FetchStruct = driver.multiEval.listPy.segment.acp.extended.standardDev.
↪fetch(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2.8 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Average_Acp: List[float]: No parameter help available
- Ms_Power_Wide: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOut: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Average_Acp: List[float]: No parameter help available
- Ms_Power_Wide: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOut: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:AVERage
value: CalculateStruct = driver.multiEval.listPy.segment.acp.average.
↳calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:AVERage
value: FetchStruct = driver.multiEval.listPy.segment.acp.average.fetch(segment.
↳= repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2.9 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:ACP:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Maximum_Acp: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Maximum_Acp: List[float]: No parameter help available
- Ms_Power_Wide: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:ACP:MAXimum
value: CalculateStruct = driver.multiEval.listPy.segment.acp.maximum.
↪calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERAGE, MINimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for

each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>
↪:ACP:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.acp.maximum.fetch(segment,
↪= repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2.10 Minimum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:ACP:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:ACP:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Minimum_Acp: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm

- **Out_Of_Tol_Count**: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count**: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- **Reliability**: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability**: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Minimum_Acp**: List[float]: No parameter help available
- **Ms_Power_Wide**: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Ms_Power_Narrow**: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm
- **Out_Of_Tol_Count**: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count**: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:MINimum
value: CalculateStruct = driver.multiEval.listPy.segment.acp.minimum.
↳calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERAGE, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:MINimum
value: FetchStruct = driver.multiEval.listPy.segment.acp.minimum.fetch(segment_
↳= repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERAGE, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.2.11 StandardDev

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:ACP:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:ACP:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Std_Dev_Acp: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFIGure:CDMA:MEASi:MEvaluation:SCount: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: float: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.

- Std_Dev_Acp: List[float]: No parameter help available
- Ms_Power_Wide: float: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: float: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42dBm , Unit: dBm
- Out_Of_Tol_Count: float: float Out of tolerance result, i.e. percentage of measurement intervals of the statistic count (CONFigure:CDMA:MEASi:MEValuation:SCOUNT: MODulation) exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: int: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:SDEviation
value: CalculateStruct = driver.multiEval.listPy.segment.acp.standardDev.
↳calculate(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:ACP:SDEviation
value: FetchStruct = driver.multiEval.listPy.segment.acp.standardDev.
↳fetch(segment = repcap.Segment.Default)
```

Returns the adjacent channel power (ACP) results for segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrow-band power results are returned, too. To define the statistical length for AVERage, MINimum, MAXimum and SDEviation calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.3 Obw

class Obw

Obw commands group definition. 8 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.segment.obw.clone()
```

Subgroups

7.4.6.3.3.1 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:OBW:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:OBW:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz
- Lower_Freq: float: float Lower frequency of the occupied bandwidth. Range: -8 MHz to 0 MHz , Unit: Hz
- Upper_Freq: float: float Upper frequency of the occupied bandwidth. Range: 0 MHz to 8 MHz , Unit: Hz

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: Hz
- Lower_Freq: float: float Lower frequency of the occupied bandwidth. Range: -8 MHz to 0 MHz , Unit: Hz

- Upper_Freq: float: float Upper frequency of the occupied bandwidth. Range: 0 MHz to 8 MHz , Unit: Hz

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>
↪:OBW:CURRENT
value: CalculateStruct = driver.multiEval.listPy.segment.obw.current.
↪calculate(segment = repcap.Segment.Default)
```

Returns CURRENT occupied bandwidth (OBW) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>
↪:OBW:CURRENT
value: FetchStruct = driver.multiEval.listPy.segment.obw.current.fetch(segment,
↪= repcap.Segment.Default)
```

Returns CURRENT occupied bandwidth (OBW) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.3.2 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:OBW:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:OBW:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.

- **Seg_Reliability**: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- **Obw**: float: float Occupied bandwidth Range: 0 MHz to 8 MHz (SDEviation 0 MHz to 4MHz) , Unit: Hz

class FetchStruct

Response structure. Fields:

- **Reliability**: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability**: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- **Obw**: float: float Occupied bandwidth Range: 0 MHz to 8 MHz (SDEviation 0 MHz to 4MHz) , Unit: Hz

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:OBW:AVERAge
value: CalculateStruct = driver.multiEval.listPy.segment.obw.average.
↳calculate(segment = repcap.Segment.Default)
```

Returns occupied bandwidth (OBW) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAge, MAXimum and SDEviation calculation and to enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:OBW:AVERAge
value: FetchStruct = driver.multiEval.listPy.segment.obw.average.fetch(segment_
↳ repcap.Segment.Default)
```

Returns occupied bandwidth (OBW) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAge, MAXimum and SDEviation calculation and to enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.3.3 Maximum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:OBW:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:OBW:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz (SDEviation 0 MHz to 4MHz) , Unit: Hz

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz (SDEviation 0 MHz to 4MHz) , Unit: Hz

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>
↳ :OBW:MAXimum
value: CalculateStruct = driver.multiEval.listPy.segment.obw.maximum.
↳ calculate(segment = repcap.Segment.Default)
```

Returns occupied bandwidth (OBW) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly

stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪ :OBW:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.obw.maximum.fetch(segment,
↪ = repcap.Segment.Default)
```

Returns occupied bandwidth (OBW) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value). To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.3.4 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:OBW:SDEViation
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:OBW:SDEViation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz (SDEviation 0 MHz to 4MHz) , Unit: Hz

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Obw: float: float Occupied bandwidth Range: 0 MHz to 8 MHz (SDEviation 0 MHz to 4MHz) , Unit: Hz

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>
↪:OBW:SDEviation
value: CalculateStruct = driver.multiEval.listPy.segment.obw.standardDev.
↪calculate(segment = repcap.Segment.Default)
```

Returns occupied bandwidth (OBW) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>
↪:OBW:SDEviation
value: FetchStruct = driver.multiEval.listPy.segment.obw.standardDev.
↪fetch(segment = repcap.Segment.Default)
```

Returns occupied bandwidth (OBW) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.4 Cp

class Cp

Cp commands group definition. 9 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.segment.cp.clone()
```

Subgroups

7.4.6.3.4.1 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CP:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CP:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rdc_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rcc_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rea_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rfch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_0_W_02_E_04: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_0_W_01_E_02: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- **Reliability**: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability**: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- **Rpi_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rdc_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rcc_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rea_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rfch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_0_W_02_E_04**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_0_W_01_E_02**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_1_W_06_E_08**: float: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:CP:CURRENT
value: CalculateStruct = driver.multiEval.listPy.segment.cp.current.
↳calculate(segment = repcap.Segment.Default)
```

Returns channel power (CP) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>:CP:CURRENT
value: FetchStruct = driver.multiEval.listPy.segment.cp.current.fetch(segment =
↳repcap.Segment.Default)
```

(continues on next page)

(continued from previous page)

Returns channel power (CP) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.4.2 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CP:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CP:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rdc_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rcc_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rea_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rfch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_0_W_02_E_04: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_0_W_01_E_02: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- **Reliability**: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability**: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- **Rpi_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rdc_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rcc_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rea_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rfch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_0_W_02_E_04**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_0_W_01_E_02**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_1_W_06_E_08**: float: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:CP:AVERage
value: CalculateStruct = driver.multiEval.listPy.segment.cp.average.
↳calculate(segment = repcap.Segment.Default)
```

Returns channel power (CP) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>:CP:AVERage
value: FetchStruct = driver.multiEval.listPy.segment.cp.average.fetch(segment =
↳repcap.Segment.Default)
```

(continues on next page)

(continued from previous page)

Returns channel power (CP) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.4.3 Maximum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CP:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CP:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rdc_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rcc_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rea_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rfch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_0_W_02_E_04: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_0_W_01_E_02: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- **Reliability**: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability**: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- **Rpi_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rdc_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rcc_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rea_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rfch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_0_W_02_E_04**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_0_W_01_E_02**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_1_W_06_E_08**: float: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:CP:MAXimum
value: CalculateStruct = driver.multiEval.listPy.segment.cp.maximum.
↳calculate(segment = repcap.Segment.Default)
```

Returns channel power (CP) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>:CP:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.cp.maximum.fetch(segment =
↳repcap.Segment.Default)
```

(continues on next page)

(continued from previous page)

Returns channel power (CP) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.4.4 Minimum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CP:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CP:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rdc_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rcc_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rea_Ch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rfch: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_0_W_02_E_04: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_0_W_01_E_02: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- **Reliability**: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability**: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- **Rpi_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rdc_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rcc_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rea_Ch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rfch**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_0_W_02_E_04**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_0_W_01_E_02**: float: float RMS channel power values for the indicated channels. Range: -25 dB to 0 dB (SDEViation 0 dB to 25 dB) Unit: dB
- **Rsch_1_W_06_E_08**: float: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:CP:MINimum
value: CalculateStruct = driver.multiEval.listPy.segment.cp.minimum.
↳calculate(segment = repcap.Segment.Default)
```

Returns channel power (CP) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>:CP:MINimum
value: FetchStruct = driver.multiEval.listPy.segment.cp.minimum.fetch(segment =
↳repcap.Segment.Default)
```

(continues on next page)

(continued from previous page)

Returns channel power (CP) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and enable the calculation of the results use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.4.5 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CP:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Seg_Reliability: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Rpi_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased INVisible: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rdc_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rcc_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rea_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rfch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rsch_0_W_02_E_04: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rsch_0_W_01_E_02: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rsch_1_W_06_E_08: enums.SigChStateA: For future use - returned value not relevant.
- Rsch_1_W_02_E_04: enums.SigChStateA: For future use - returned value not relevant.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>:CP:STATE
value: FetchStruct = driver.multiEval.listPy.segment.cp.state.fetch(segment = ↪
↪repcap.Segment.Default)
```

Return the states of the channels for power measurement (CP) .

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.5 Cpo

class Cpo

Cpo commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.segment.cpo.clone()
```

Subgroups

7.4.6.3.5.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CPO:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CPO:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rfch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_02_E_04: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg

- Rsch_0_W_01_E_02: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rfch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_02_E_04: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_01_E_02: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:CPO:CURRent
value: CalculateStruct = driver.multiEval.listPy.segment.cpo.current.
↪calculate(segment = repcap.Segment.Default)
```

Returns channel phase offset (CPO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:CPO:CURRent
value: FetchStruct = driver.multiEval.listPy.segment.cpo.current.fetch(segment,
↪= repcap.Segment.Default)
```

Returns channel phase offset (CPO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.5.2 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CPO:AVERAge
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CPO:AVERAge
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rfch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg

- Rsch_0_W_02_E_04: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_01_E_02: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rfch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_02_E_04: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_01_E_02: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGMENT<nr>
↳:CPO:AVERage
value: CalculateStruct = driver.multiEval.listPy.segment.cpo.average.
↳calculate(segment = repcap.Segment.Default)
```

Returns channel phase offset (CPO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪ :CPO:AVERage
value: FetchStruct = driver.multiEval.listPy.segment.cpo.average.fetch(segment,
↪ = repcap.Segment.Default)
```

Returns channel phase offset (CPO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.5.3 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CPO:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CPO:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg

- Rfch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_02_E_04: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_01_E_02: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rfch: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_02_E_04: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_01_E_02: float: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↪:CPO:MAXimum
value: CalculateStruct = driver.multiEval.listPy.segment.cpo.maximum.
↪calculate(segment = repcap.Segment.Default)
```

Returns channel phase offset (CPO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for

each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>
↪:CPO:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.cpo.maximum.fetch(segment,
↪= repcap.Segment.Default)
```

Returns channel phase offset (CPO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.5.4 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CPO:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Seg_Reliability: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Rpi_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rdc_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rcc_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rea_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rfch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rsch_0_W_02_E_04: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased

- Rsch_0_W_01_E_02: enums.SigChStateA: INVisible | ACTive | IACTive | ALLased
- Rsch_1_W_06_E_08: enums.SigChStateA: For future use - returned value not relevant.
- Rsch_1_W_02_E_04: enums.SigChStateA: For future use - returned value not relevant.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>:CPO:STATe
value: FetchStruct = driver.multiEval.listPy.segment.cpo.state.fetch(segment =
↳repcap.Segment.Default)
```

Return the states of the channels for channel phase offset (CPO) measurements.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.6 Cto

class Cto

Cto commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.segment.cto.clone()
```

Subgroups

7.4.6.3.6.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CTO:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CTO:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s

- Rdc_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rcc_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rea_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rfch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_02_E_04: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_01_E_02: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rdc_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rcc_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rea_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rfch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_02_E_04: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_01_E_02: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:CTO:CURRENT
value: CalculateStruct = driver.multiEval.listPy.segment.cto.current.
↳calculate(segment = repcap.Segment.Default)
```

Returns channel time offset (CTO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value). To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:CTO:CURRENT
value: FetchStruct = driver.multiEval.listPy.segment.cto.current.fetch(segment,
↳= repcap.Segment.Default)
```

Returns channel time offset (CTO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value). To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface 'Segment')

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.6.2 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CTO:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CTO:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.

- **Seg_Reliability:** int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- **Rpi_Ch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rdc_Ch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rcc_Ch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rea_Ch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rfch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_0_W_02_E_04:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_0_W_01_E_02:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_1_W_06_E_08:** float: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04:** float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- **Reliability:** int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability:** int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- **Rpi_Ch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rdc_Ch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rcc_Ch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rea_Ch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rfch:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_0_W_02_E_04:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_0_W_01_E_02:** float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_1_W_06_E_08:** float: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04:** float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:CTO:AVERage
value: CalculateStruct = driver.multiEval.listPy.segment.cto.average.
↳calculate(segment = repcap.Segment.Default)
```

Returns channel time offset (CTO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>
↳:CTO:AVERage
value: FetchStruct = driver.multiEval.listPy.segment.cto.average.fetch(segment,
↳= repcap.Segment.Default)
```

Returns channel time offset (CTO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.6.3 Maximum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CTO:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CTO:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rdc_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rcc_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rea_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rfch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_02_E_04: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_01_E_02: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: int: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error. If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rdc_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rcc_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rea_Ch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rfch: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_02_E_04: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_01_E_02: float: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s

- Rsch_1_W_06_E_08: float: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: float: float For future use - returned value not relevant.

calculate(segment=<Segment.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>
↳:CTO:MAXimum
value: CalculateStruct = driver.multiEval.listPy.segment.cto.maximum.
↳calculate(segment = repcap.Segment.Default)
```

Returns channel time offset (CTO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<nr>
↳:CTO:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.cto.maximum.fetch(segment_
↳= repcap.Segment.Default)
```

Returns channel time offset (CTO) results for the segment <no> in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.3.6.4 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CTO:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Seg_Reliability: int: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Rpi_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rdc_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rcc_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rea_Ch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rfch: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rsch_0_W_02_E_04: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rsch_0_W_01_E_02: enums.SigChStateA: INVisible | ACTive | IACTive | ALIased
- Rsch_1_W_06_E_08: enums.SigChStateA: For future use - returned value not relevant.
- Rsch_1_W_02_E_04: enums.SigChStateA: For future use - returned value not relevant.

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>:CTO:STATe
value: FetchStruct = driver.multiEval.listPy.segment.cto.state.fetch(segment = repcap
↪repcap.Segment.Default)
```

Return the states of the channels for channel time offset (CTO) measurements.

param segment optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Segment’)

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4 Acp

class Acp

Acp commands group definition. 66 total commands, 12 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.acp.clone()
```

Subgroups

7.4.6.4.1 Otolerance

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:OTOLerance
```

class Otolerance

Otolerance commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[int]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:OTOLerance
value: List[int] = driver.multiEval.listPy.acp.otolerance.fetch()
```

Returns the out of tolerance percentages in the adjacent channel power measurement for all active list mode segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return out_of_tol_count: decimal The percentage of measurement intervals of the statistic count exceeding the specified limits. Comma-separated list of values, one per active segment. Range: 0 % to 100 % , Unit: %

7.4.6.4.2 StCount

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:STCount
```

class StCount

StCount commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[int]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:STCount
value: List[int] = driver.multiEval.listPy.acp.stCount.fetch()
```

Returns the statistic count of the spectrum measurement for all active list mode segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return statistic_count: decimal Number of evaluated valid slots. Comma-separated list of values, one per active segment. Range: 0 to 1000

7.4.6.4.3 Acpm<AcpMinus>

RepCap Settings

```
# Range: Ch1 .. Ch20
rc = driver.multiEval.listPy.acp.acpm.repcap_acpMinus_get()
driver.multiEval.listPy.acp.acpm.repcap_acpMinus_set(repcap.AcpMinus.Ch1)
```

class Acpm

Acpm commands group definition. 6 total commands, 3 Sub-groups, 0 group commands Repeated Capability: AcpMinus, default value after init: AcpMinus.Ch1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.acp.acpm.clone()
```

Subgroups

7.4.6.4.3.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM<AcpMinus>:CURRENT
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM<AcpMinus>:CURRENT
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(acpMinus=<AcpMinus.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM
↪<freqpoint>:CURRENT
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.acpm.current.
↪calculate(acpMinus = repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels). The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

fetch(acpMinus=<AcpMinus.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPM<freqpoint>
↳:CURRent
value: List[float] = driver.multiEval.listPy.acp.acpm.current.fetch(acpMinus =
↳repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

7.4.6.4.3.2 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPM<AcpMinus>:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPM<AcpMinus>:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(acpMinus=<AcpMinus.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPM
↳<freqpoint>:AVERage
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.acpm.average.
↳calculate(acpMinus = repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

fetch(acpMinus=<AcpMinus.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPM<freqpoint>
↳:AVERage
value: List[float] = driver.multiEval.listPy.acp.acpm.average.fetch(acpMinus =
↳repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels) . The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

7.4.6.4.3.3 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPM<AcpMinus>:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPM<AcpMinus>:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(acpMinus=<AcpMinus.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPM
↪<freqpoint>:MAXimum
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.acpm.maximum.
↪calculate(acpMinus = repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels) . The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

fetch(acpMinus=<AcpMinus.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPM<freqpoint>
↪:MAXimum
value: List[float] = driver.multiEval.listPy.acp.acpm.maximum.fetch(acpMinus =
↪repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels) . The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface 'Acpm')

return acpm: No help available

7.4.6.4.4 Extended

class Extended

Extended commands group definition. 22 total commands, 7 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.acp.extended.clone()
```

Subgroups

7.4.6.4.4.1 Acpm<AcpMinus>

RepCap Settings

```
# Range: Ch1 .. Ch20
rc = driver.multiEval.listPy.acp.extended.acpm.repcap_acpMinus_get()
driver.multiEval.listPy.acp.extended.acpm.repcap_acpMinus_set(repcap.AcpMinus.Ch1)
```

class Acpm

Acpm commands group definition. 6 total commands, 3 Sub-groups, 0 group commands Repeated Capability: AcpMinus, default value after init: AcpMinus.Ch1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.acp.extended.acpm.clone()
```

Subgroups

7.4.6.4.4.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM<AcpMinus>:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM<AcpMinus>:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(*acpMinus*=<*AcpMinus.Default*: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM
↪<freqpoint>:CURRENT
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.extended.acpm.
↪current.calculate(acpMinus = repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

fetch(*acpMinus*=<*AcpMinus.Default*: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM
↪<freqpoint>:CURRENT
value: List[float] = driver.multiEval.listPy.acp.extended.acpm.current.
↪fetch(acpMinus = repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

7.4.6.4.4.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM<AcpMinus>:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM<AcpMinus>:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(*acpMinus*=<*AcpMinus.Default*: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]


```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM
↪<freqpoint>:AVERage
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.extended.acpm.
↪average.calculate(acpMinus = repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

fetch(acpMinus=<AcpMinus.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM
↪<freqpoint>:AVERage
value: List[float] = driver.multiEval.listPy.acp.extended.acpm.average.
↪fetch(acpMinus = repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

7.4.6.4.4 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM<AcpMinus>:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM<AcpMinus>:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(acpMinus=<AcpMinus.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM
↪<freqpoint>:MAXimum
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.extended.acpm.
↪maximum.calculate(acpMinus = repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels) . The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

fetch(acpMinus=<AcpMinus.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPM
↪<freqpoint>:MAXimum
value: List[float] = driver.multiEval.listPy.acp.extended.acpm.maximum.
↪fetch(acpMinus = repcap.AcpMinus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels) . The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpMinus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpm’)

return acpm: No help available

7.4.6.4.4.5 Acpp<AcpPlus>

RepCap Settings

```
# Range: Ch1 .. Ch20
rc = driver.multiEval.listPy.acp.extended.acpp.repcap_acpPlus_get()
driver.multiEval.listPy.acp.extended.acpp.repcap_acpPlus_set(repcap.AcpPlus.Ch1)
```

class Acpp

Acpp commands group definition. 6 total commands, 3 Sub-groups, 0 group commands Repeated Capability: AcpPlus, default value after init: AcpPlus.Ch1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.acp.extended.acpp.clone()
```

Subgroups

7.4.6.4.4.6 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP<AcpPlus>:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP<AcpPlus>:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(acpPlus=<AcpPlus.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP
↳<freqpoint>:CURRent
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.extended.acpp.
↳current.calculate(acpPlus = repcap.AcpPlus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpp’)

return acpm: No help available

fetch(acpPlus=<AcpPlus.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP
↳<freqpoint>:CURRent
value: List[float] = driver.multiEval.listPy.acp.extended.acpp.current.
↳fetch(acpPlus = repcap.AcpPlus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpp’)

return acpm: No help available

7.4.6.4.4.7 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP<AcpPlus>:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP<AcpPlus>:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(acpPlus=<AcpPlus.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP
↪<freqpoint>:AVERage
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.extended.acpp.
↪average.calculate(acpPlus = repcap.AcpPlus.Default)

```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acp’)

return acpm: No help available

fetch(acpPlus=<AcpPlus.Default: -1>) → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP
↪<freqpoint>:AVERage
value: List[float] = driver.multiEval.listPy.acp.extended.acpp.average.
↪fetch(acpPlus = repcap.AcpPlus.Default)

```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acp’)

return acpm: No help available

7.4.6.4.4.8 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP<AcpPlus>:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP<AcpPlus>:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(*acpPlus*=<*AcpPlus.Default*: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP
↪<freqpoint>:MAXimum
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.extended.acpp.
↪maximum.calculate(acpPlus = repcap.AcpPlus.Default)

```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acp’)

return acpm: No help available

fetch(*acpPlus*=<*AcpPlus.Default*: -1>) → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:ACPP
↪<freqpoint>:MAXimum
value: List[float] = driver.multiEval.listPy.acp.extended.acpp.maximum.
↪fetch(acpPlus = repcap.AcpPlus.Default)

```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 20 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acp’)

return acpm: No help available

7.4.6.4.4.9 Current

SCPI Commands

```
FEtCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTEnded:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTEnded:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acp: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acp: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:ACP:EXTended:CURRENT
value: CalculateStruct = driver.multiEval.listPy.acp.extended.current.
↳calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:CURRENT
value: FetchStruct = driver.multiEval.listPy.acp.extended.current.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4.4.10 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.

- **Seg_Reliability:** List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Acp:** List[float]: No parameter help available
- **Ms_Power_Wide:** List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Ms_Power_Narrow:** List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Out_Of_Tol_Count:** List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- **Reliability:** int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability:** List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Acp:** List[float]: No parameter help available
- **Ms_Power_Wide:** List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Ms_Power_Narrow:** List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Out_Of_Tol_Count:** List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:ACP:EXTended:AVERage
value: CalculateStruct = driver.multiEval.listPy.acp.extended.average.
↳calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method `RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value`) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method `RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set`. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method `RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count`. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:AVERage
value: FetchStruct = driver.multiEval.listPy.acp.extended.average.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4.4.11 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:EXTended:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acp: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acp: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:ACP:EXTended:MAXimum
value: CalculateStruct = driver.multiEval.listPy.acp.extended.maximum.
↳calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: { ... }seg 1, { ... }seg 2, ..., { ... }seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:MAXimum
value: FetchStruct = driver.multiEval.listPy.acp.extended.maximum.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: { ... }seg 1, { ... }seg 2, ..., { ... }seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4.4.12 Minimum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acp: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acp: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:ACP:EXTended:MINimum
value: CalculateStruct = driver.multiEval.listPy.acp.extended.minimum.
↳calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:MINimum
value: FetchStruct = driver.multiEval.listPy.acp.extended.minimum.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4.4.13 StandardDev

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.

- **Seg_Reliability:** List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Acp:** List[float]: No parameter help available
- **Ms_Power_Wide:** List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Ms_Power_Narrow:** List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Out_Of_Tol_Count:** List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- **Reliability:** int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability:** List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Acp:** List[float]: No parameter help available
- **Ms_Power_Wide:** List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Ms_Power_Narrow:** List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Out_Of_Tol_Count:** List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:ACP:EXTended:SDEviation
value: CalculateStruct = driver.multiEval.listPy.acp.extended.standardDev.
↳calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method `RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value`) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method `RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set`. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method `RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count`. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:ACP:EXTended:SDEviation
value: FetchStruct = driver.multiEval.listPy.acp.extended.standardDev.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value). If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4.5 Acpp<AcpPlus>

RepCap Settings

```
# Range: Ch1 .. Ch20
rc = driver.multiEval.listPy.acp.acpp.repcap_acpPlus_get()
driver.multiEval.listPy.acp.acpp.repcap_acpPlus_set(repcap.AcpPlus.Ch1)
```

class Acpp

Acpp commands group definition. 6 total commands, 3 Sub-groups, 0 group commands Repeated Capability: AcpPlus, default value after init: AcpPlus.Ch1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.acp.acpp.clone()
```

Subgroups

7.4.6.4.5.1 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPP<AcpPlus>:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPP<AcpPlus>:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(acpPlus=<AcpPlus.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP
↪<freqpoint>:CURRent
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.acpp.current.
↪calculate(acpPlus = repcap.AcpPlus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpp’)

return acpm: No help available

fetch(acpPlus=<AcpPlus.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP<freqpoint>
↪:CURRent
value: List[float] = driver.multiEval.listPy.acp.acpp.current.fetch(acpPlus = ↪
↪repcap.AcpPlus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpp’)

return acpm: No help available

7.4.6.4.5.2 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP<AcpPlus>:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP<AcpPlus>:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(acpPlus=<AcpPlus.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP
↪<freqpoint>:AVERage
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.acpp.average.
↪calculate(acpPlus = repcap.AcpPlus.Default)
```


Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpp’)

return acpm: No help available

fetch(*acpPlus*=<*Acplus.Default*: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP<freqpoint>
↳:AVERage
value: List[float] = driver.multiEval.listPy.acp.acpp.average.fetch(acpPlus =
↳repcap.AcpPlus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface ‘Acpp’)

return acpm: No help available

7.4.6.4.5.3 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP<AcpPlus>:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP<AcpPlus>:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate(*acpPlus*=<*Acplus.Default*: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP
↳<freqpoint>:MAXimum
value: List[enums.ResultStatus2] = driver.multiEval.listPy.acp.acpp.maximum.
↳calculate(acpPlus = repcap.AcpPlus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the ‘minus’ direction, the mnemonic ACPP to the channels in the ‘plus’ direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface 'Acpp')

return acpm: No help available

fetch(*acpPlus*=<AcpPlus.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:ACPP<freqpoint>
↪:MAXimum
value: List[float] = driver.multiEval.listPy.acp.acpp.maximum.fetch(acpPlus = ↪
↪repcap.AcpPlus.Default)
```

Returns adjacent channel power (statistical) values for the selected off-center channels and all active list mode segments. The mnemonic ACPM refers to the channels in the 'minus' direction, the mnemonic ACPP to the channels in the 'plus' direction relative to the current channel. The offset is selected with the <freqpoint> suffix (± 10 channels). The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param acpPlus optional repeated capability selector. Default value: Ch1 (settable in the interface 'Acpp')

return acpm: No help available

7.4.6.4.6 Npow

class Npow

Npow commands group definition. 10 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.acp.npow.clone()
```

Subgroups

7.4.6.4.6.1 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:CURRENT
value: List[float] = driver.multiEval.listPy.acp.npow.current.calculate()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:CURRENT
value: List[float] = driver.multiEval.listPy.acp.npow.current.fetch()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.6.2 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:AVERage
value: List[float] = driver.multiEval.listPy.acp.npow.average.calculate()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:AVERage
value: List[float] = driver.multiEval.listPy.acp.npow.average.fetch()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.6.3 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:MAXimum
value: List[float] = driver.multiEval.listPy.acp.npow.maximum.calculate()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:NPOW:MAXimum
value: List[float] = driver.multiEval.listPy.acp.npow.maximum.fetch()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.6.4 Minimum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:NPOW:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:NPOW:MINimum

```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:NPOW:MINimum
value: List[float] = driver.multiEval.listPy.acp.npow.minimum.calculate()

```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:NPOW:MINimum
value: List[float] = driver.multiEval.listPy.acp.npow.minimum.fetch()

```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.6.5 StandardDev

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:NPOW:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:NPOW:SDEviation

```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:ACP:NPOW:SDEVIation
value: List[float] = driver.multiEval.listPy.acp.npow.standardDev.calculate()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:NPOW:SDEVIation
value: List[float] = driver.multiEval.listPy.acp.npow.standardDev.fetch()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return narrow_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.7 Wpow

class Wpow

Wpow commands group definition. 10 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.acp.wpow.clone()
```

Subgroups

7.4.6.4.7.1 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:WPOW:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:WPOW:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:CURRent
value: List[float] = driver.multiEval.listPy.acp.wpow.current.calculate()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:CURRent
value: List[float] = driver.multiEval.listPy.acp.wpow.current.fetch()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.7.2 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:AVERage
value: List[float] = driver.multiEval.listPy.acp.wpow.average.calculate()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:AVERage
value: List[float] = driver.multiEval.listPy.acp.wpow.average.fetch()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.7.3 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:MAXimum
value: List[float] = driver.multiEval.listPy.acp.wpow.maximum.calculate()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:MAXimum
value: List[float] = driver.multiEval.listPy.acp.wpow.maximum.fetch()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.7.4 Minimum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:MINimum

```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:MINimum
value: List[float] = driver.multiEval.listPy.acp.wpow.minimum.calculate()

```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:MINimum
value: List[float] = driver.multiEval.listPy.acp.wpow.minimum.fetch()

```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.7.5 StandardDev

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:SDEVIation
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:WPOW:SDEVIation

```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]


```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:ACP:WPOW:SDEviation
value: List[float] = driver.multiEval.listPy.acp.wpow.standardDev.calculate()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:WPOW:SDEviation
value: List[float] = driver.multiEval.listPy.acp.wpow.standardDev.fetch()
```

Returns narrowband and wideband (statistical) power values in the ACP measurement for all active list mode segments. The narrowband filter is 1.23 MHz, the wideband filter is 8 MHz wide. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return wideband_power: float Comma-separated list of values, one per active segment.
Range: -55 dBm to 42 dBm

7.4.6.4.8 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acpm_10: List[enums.ResultStatus2]: No parameter help available
- Acpm_9: List[enums.ResultStatus2]: No parameter help available
- Acpm_8: List[enums.ResultStatus2]: No parameter help available
- Acpm_7: List[enums.ResultStatus2]: No parameter help available

- Acpm_6: List[enums.ResultStatus2]: No parameter help available
- Acpm_5: List[enums.ResultStatus2]: No parameter help available
- Acpm_4: List[enums.ResultStatus2]: No parameter help available
- Acpm_3: List[enums.ResultStatus2]: No parameter help available
- Acpm_2: List[enums.ResultStatus2]: No parameter help available
- Acpm_1: List[enums.ResultStatus2]: No parameter help available
- Acp_Carrier: List[enums.ResultStatus2]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB
- Acpp_1: List[enums.ResultStatus2]: No parameter help available
- Acpp_2: List[enums.ResultStatus2]: No parameter help available
- Acpp_3: List[enums.ResultStatus2]: No parameter help available
- Acpp_4: List[enums.ResultStatus2]: No parameter help available
- Acpp_5: List[enums.ResultStatus2]: No parameter help available
- Acpp_6: List[enums.ResultStatus2]: No parameter help available
- Acpp_7: List[enums.ResultStatus2]: No parameter help available
- Acpp_8: List[enums.ResultStatus2]: No parameter help available
- Acpp_9: List[enums.ResultStatus2]: No parameter help available
- Acpp_10: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acpm_10: List[float]: No parameter help available
- Acpm_9: List[float]: No parameter help available
- Acpm_8: List[float]: No parameter help available
- Acpm_7: List[float]: No parameter help available
- Acpm_6: List[float]: No parameter help available

- Acpm_5: List[float]: No parameter help available
- Acpm_4: List[float]: No parameter help available
- Acpm_3: List[float]: No parameter help available
- Acpm_2: List[float]: No parameter help available
- Acpm_1: List[float]: No parameter help available
- Acp_Carrier: List[float]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB
- Acpp_1: List[float]: No parameter help available
- Acpp_2: List[float]: No parameter help available
- Acpp_3: List[float]: No parameter help available
- Acpp_4: List[float]: No parameter help available
- Acpp_5: List[float]: No parameter help available
- Acpp_6: List[float]: No parameter help available
- Acpp_7: List[float]: No parameter help available
- Acpp_8: List[float]: No parameter help available
- Acpp_9: List[float]: No parameter help available
- Acpp_10: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:CURRent
value: CalculateStruct = driver.multiEval.listPy.acp.current.calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERAGE, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:CURRent
value: FetchStruct = driver.multiEval.listPy.acp.current.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4.9 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acpm_10: List[enums.ResultStatus2]: No parameter help available
- Acpm_9: List[enums.ResultStatus2]: No parameter help available
- Acpm_8: List[enums.ResultStatus2]: No parameter help available
- Acpm_7: List[enums.ResultStatus2]: No parameter help available
- Acpm_6: List[enums.ResultStatus2]: No parameter help available
- Acpm_5: List[enums.ResultStatus2]: No parameter help available
- Acpm_4: List[enums.ResultStatus2]: No parameter help available
- Acpm_3: List[enums.ResultStatus2]: No parameter help available
- Acpm_2: List[enums.ResultStatus2]: No parameter help available
- Acpm_1: List[enums.ResultStatus2]: No parameter help available
- Acp_Carrier: List[enums.ResultStatus2]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB

- Acpp_1: List[enums.ResultStatus2]: No parameter help available
- Acpp_2: List[enums.ResultStatus2]: No parameter help available
- Acpp_3: List[enums.ResultStatus2]: No parameter help available
- Acpp_4: List[enums.ResultStatus2]: No parameter help available
- Acpp_5: List[enums.ResultStatus2]: No parameter help available
- Acpp_6: List[enums.ResultStatus2]: No parameter help available
- Acpp_7: List[enums.ResultStatus2]: No parameter help available
- Acpp_8: List[enums.ResultStatus2]: No parameter help available
- Acpp_9: List[enums.ResultStatus2]: No parameter help available
- Acpp_10: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acpm_10: List[float]: No parameter help available
- Acpm_9: List[float]: No parameter help available
- Acpm_8: List[float]: No parameter help available
- Acpm_7: List[float]: No parameter help available
- Acpm_6: List[float]: No parameter help available
- Acpm_5: List[float]: No parameter help available
- Acpm_4: List[float]: No parameter help available
- Acpm_3: List[float]: No parameter help available
- Acpm_2: List[float]: No parameter help available
- Acpm_1: List[float]: No parameter help available
- Acp_Carrier: List[float]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB
- Acpp_1: List[float]: No parameter help available

- Acpp_2: List[float]: No parameter help available
- Acpp_3: List[float]: No parameter help available
- Acpp_4: List[float]: No parameter help available
- Acpp_5: List[float]: No parameter help available
- Acpp_6: List[float]: No parameter help available
- Acpp_7: List[float]: No parameter help available
- Acpp_8: List[float]: No parameter help available
- Acpp_9: List[float]: No parameter help available
- Acpp_10: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:AVERage
value: CalculateStruct = driver.multiEval.listPy.acp.average.calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:AVERage
value: FetchStruct = driver.multiEval.listPy.acp.average.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by

method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4.10 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acpm_10: List[enums.ResultStatus2]: No parameter help available
- Acpm_9: List[enums.ResultStatus2]: No parameter help available
- Acpm_8: List[enums.ResultStatus2]: No parameter help available
- Acpm_7: List[enums.ResultStatus2]: No parameter help available
- Acpm_6: List[enums.ResultStatus2]: No parameter help available
- Acpm_5: List[enums.ResultStatus2]: No parameter help available
- Acpm_4: List[enums.ResultStatus2]: No parameter help available
- Acpm_3: List[enums.ResultStatus2]: No parameter help available
- Acpm_2: List[enums.ResultStatus2]: No parameter help available
- Acpm_1: List[enums.ResultStatus2]: No parameter help available
- Acp_Carrier: List[enums.ResultStatus2]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB
- Acpp_1: List[enums.ResultStatus2]: No parameter help available
- Acpp_2: List[enums.ResultStatus2]: No parameter help available
- Acpp_3: List[enums.ResultStatus2]: No parameter help available
- Acpp_4: List[enums.ResultStatus2]: No parameter help available
- Acpp_5: List[enums.ResultStatus2]: No parameter help available
- Acpp_6: List[enums.ResultStatus2]: No parameter help available
- Acpp_7: List[enums.ResultStatus2]: No parameter help available

- Acpp_8: List[enums.ResultStatus2]: No parameter help available
- Acpp_9: List[enums.ResultStatus2]: No parameter help available
- Acpp_10: List[enums.ResultStatus2]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acpm_10: List[float]: No parameter help available
- Acpm_9: List[float]: No parameter help available
- Acpm_8: List[float]: No parameter help available
- Acpm_7: List[float]: No parameter help available
- Acpm_6: List[float]: No parameter help available
- Acpm_5: List[float]: No parameter help available
- Acpm_4: List[float]: No parameter help available
- Acpm_3: List[float]: No parameter help available
- Acpm_2: List[float]: No parameter help available
- Acpm_1: List[float]: No parameter help available
- Acp_Carrier: List[float]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB
- Acpp_1: List[float]: No parameter help available
- Acpp_2: List[float]: No parameter help available
- Acpp_3: List[float]: No parameter help available
- Acpp_4: List[float]: No parameter help available
- Acpp_5: List[float]: No parameter help available
- Acpp_6: List[float]: No parameter help available
- Acpp_7: List[float]: No parameter help available
- Acpp_8: List[float]: No parameter help available

- Acpp_9: List[float]: No parameter help available
- Acpp_10: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:MAXimum
value: CalculateStruct = driver.multiEval.listPy.acp.maximum.calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:MAXimum
value: FetchStruct = driver.multiEval.listPy.acp.maximum.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4.11 Minimum

SCPI Commands

<code>FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:MINimum</code> <code>CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:MINimum</code>
--

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acpm_10: List[float]: No parameter help available
- Acpm_9: List[float]: No parameter help available
- Acpm_8: List[float]: No parameter help available
- Acpm_7: List[float]: No parameter help available
- Acpm_6: List[float]: No parameter help available
- Acpm_5: List[float]: No parameter help available
- Acpm_4: List[float]: No parameter help available
- Acpm_3: List[float]: No parameter help available
- Acpm_2: List[float]: No parameter help available
- Acpm_1: List[float]: No parameter help available
- Acp_Carrier: List[float]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB
- Acpp_1: List[float]: No parameter help available
- Acpp_2: List[float]: No parameter help available
- Acpp_3: List[float]: No parameter help available
- Acpp_4: List[float]: No parameter help available
- Acpp_5: List[float]: No parameter help available
- Acpp_6: List[float]: No parameter help available
- Acpp_7: List[float]: No parameter help available
- Acpp_8: List[float]: No parameter help available
- Acpp_9: List[float]: No parameter help available
- Acpp_10: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm

- **Ms_Power_Narrow:** List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Out_Of_Tol_Count:** List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count:** List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- **Reliability:** int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- **Seg_Reliability:** List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- **Acpm_10:** List[float]: No parameter help available
- **Acpm_9:** List[float]: No parameter help available
- **Acpm_8:** List[float]: No parameter help available
- **Acpm_7:** List[float]: No parameter help available
- **Acpm_6:** List[float]: No parameter help available
- **Acpm_5:** List[float]: No parameter help available
- **Acpm_4:** List[float]: No parameter help available
- **Acpm_3:** List[float]: No parameter help available
- **Acpm_2:** List[float]: No parameter help available
- **Acpm_1:** List[float]: No parameter help available
- **Acp_Carrier:** List[float]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB
- **Acpp_1:** List[float]: No parameter help available
- **Acpp_2:** List[float]: No parameter help available
- **Acpp_3:** List[float]: No parameter help available
- **Acpp_4:** List[float]: No parameter help available
- **Acpp_5:** List[float]: No parameter help available
- **Acpp_6:** List[float]: No parameter help available
- **Acpp_7:** List[float]: No parameter help available
- **Acpp_8:** List[float]: No parameter help available
- **Acpp_9:** List[float]: No parameter help available
- **Acpp_10:** List[float]: No parameter help available
- **Ms_Power_Wide:** List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- **Ms_Power_Narrow:** List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm

- **Out_Of_Tol_Count**: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- **Cur_Stat_Count**: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:MINimum
value: CalculateStruct = driver.multiEval.listPy.acp.minimum.calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:MINimum
value: FetchStruct = driver.multiEval.listPy.acp.minimum.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.4.12 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acpm_10: List[float]: No parameter help available
- Acpm_9: List[float]: No parameter help available
- Acpm_8: List[float]: No parameter help available
- Acpm_7: List[float]: No parameter help available
- Acpm_6: List[float]: No parameter help available
- Acpm_5: List[float]: No parameter help available
- Acpm_4: List[float]: No parameter help available
- Acpm_3: List[float]: No parameter help available
- Acpm_2: List[float]: No parameter help available
- Acpm_1: List[float]: No parameter help available
- Acp_Carrier: List[float]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB
- Acpp_1: List[float]: No parameter help available
- Acpp_2: List[float]: No parameter help available
- Acpp_3: List[float]: No parameter help available
- Acpp_4: List[float]: No parameter help available
- Acpp_5: List[float]: No parameter help available
- Acpp_6: List[float]: No parameter help available
- Acpp_7: List[float]: No parameter help available
- Acpp_8: List[float]: No parameter help available
- Acpp_9: List[float]: No parameter help available
- Acpp_10: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[float]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Acpm_10: List[float]: No parameter help available
- Acpm_9: List[float]: No parameter help available
- Acpm_8: List[float]: No parameter help available
- Acpm_7: List[float]: No parameter help available
- Acpm_6: List[float]: No parameter help available
- Acpm_5: List[float]: No parameter help available
- Acpm_4: List[float]: No parameter help available
- Acpm_3: List[float]: No parameter help available
- Acpm_2: List[float]: No parameter help available
- Acpm_1: List[float]: No parameter help available
- Acp_Carrier: List[float]: float ACP M/P n refers to the average channel power at the carrier frequency minus/plus the frequency offset value number n. Range: -100 dB to 50 dB , Unit: dB
- Acpp_1: List[float]: No parameter help available
- Acpp_2: List[float]: No parameter help available
- Acpp_3: List[float]: No parameter help available
- Acpp_4: List[float]: No parameter help available
- Acpp_5: List[float]: No parameter help available
- Acpp_6: List[float]: No parameter help available
- Acpp_7: List[float]: No parameter help available
- Acpp_8: List[float]: No parameter help available
- Acpp_9: List[float]: No parameter help available
- Acpp_10: List[float]: No parameter help available
- Ms_Power_Wide: List[float]: float Results of the wideband (8 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Ms_Power_Narrow: List[float]: float Results of the narrowband (1.23 MHz) power measurement. Range: -55 dBm to 42 dBm , Unit: dBm
- Out_Of_Tol_Count: List[float]: decimal Out of tolerance result, i.e. percentage of measurement intervals of the statistic count exceeding the specified limits. Range: 0 % to 100 % , Unit: %
- Cur_Stat_Count: List[int]: decimal Number of evaluated valid slots in this segment. Range: 0 to 1000

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:SDEviation
value: CalculateStruct = driver.multiEval.listPy.acp.standardDev.calculate()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:ACP:SDEviation
value: FetchStruct = driver.multiEval.listPy.acp.standardDev.fetch()
```

Returns the adjacent channel power (ACP) results of each active segment (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . If enabled the wideband power and narrowband power results are returned, too. To define the statistical length for AVERage, SDEviation, MINimum and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.5 Obw

class Obw

Obw commands group definition. 18 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.obw.clone()
```

Subgroups

7.4.6.5.1 Frequency

class Frequency

Frequency commands group definition. 10 total commands, 6 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.obw.frequency.clone()
```

Subgroups

7.4.6.5.1.1 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREquency:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREquency:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:OBW:FREquency:CURRent
value: List[float] = driver.multiEval.listPy.obw.frequency.current.calculate()
```

Returns the occupied bandwidth (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per active segment. Range: 0 MHz to 8MHz (SDEViation 0 MHz to 4 MHz) , Unit: Hz

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREquency:CURRent
value: List[float] = driver.multiEval.listPy.obw.frequency.current.fetch()
```

Returns the occupied bandwidth (statistical) values for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per active segment. Range: 0 MHz to 8MHz (SDEViation 0 MHz to 4 MHz) , Unit: Hz

7.4.6.5.1.2 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:FREQuency:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:FREQuency:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
→ :MEValuation:LIST:OBW:FREQuency:AVERage
value: List[float] = driver.multiEval.listPy.obw.frequency.average.calculate()

```

Returns the occupied bandwidth (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per active segment. Range: 0 MHz to 8MHz (SDEViation 0 MHz to 4 MHz) , Unit: Hz

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:FREQuency:AVERage
value: List[float] = driver.multiEval.listPy.obw.frequency.average.fetch()

```

Returns the occupied bandwidth (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per active segment. Range: 0 MHz to 8MHz (SDEViation 0 MHz to 4 MHz) , Unit: Hz

7.4.6.5.1.3 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:FREQuency:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:FREQuency:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:OBW:FREQuency:MAXimum
value: List[float] = driver.multiEval.listPy.obw.frequency.maximum.calculate()
```

Returns the occupied bandwidth (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per active segment. Range: 0 MHz to 8MHz (SDEViation 0 MHz to 4 MHz) , Unit: Hz

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREQuency:MAXimum
value: List[float] = driver.multiEval.listPy.obw.frequency.maximum.fetch()
```

Returns the occupied bandwidth (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per active segment. Range: 0 MHz to 8MHz (SDEViation 0 MHz to 4 MHz) , Unit: Hz

7.4.6.5.1.4 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREQuency:SDEViation
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREQuency:SDEViation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:OBW:FREQuency:SDEViation
value: List[float] = driver.multiEval.listPy.obw.frequency.standardDev.
↪calculate()
```

Returns the occupied bandwidth (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per active segment. Range: 0 MHz to 8MHz (SDEViation 0 MHz to 4 MHz) , Unit: Hz

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:OBW:FREquency:SDEviation
value: List[float] = driver.multiEval.listPy.obw.frequency.standardDev.fetch()
```

Returns the occupied bandwidth (statistical) values for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per active segment. Range: 0 MHz to 8MHz (SDEviation 0 MHz to 4 MHz) , Unit: Hz

7.4.6.5.1.5 Lower

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREquency:LOWer
```

class Lower

Lower commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREquency:LOWer
value: List[float] = driver.multiEval.listPy.obw.frequency.lower.fetch()
```

Returns lower and upper OBW frequencies for all active list mode segments. The values are taken from the 'CURRent' measurement. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw_lower: float Comma-separated list of values, one per active segment. Range: 0 Hz to 100 Hz, Unit: Hz

7.4.6.5.1.6 Upper

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREquency:UPPer
```

class Upper

Upper commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:FREquency:UPPer
value: List[float] = driver.multiEval.listPy.obw.frequency.upper.fetch()
```

Returns lower and upper OBW frequencies for all active list mode segments. The values are taken from the 'CURRENT' measurement. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return obw_upper: float Comma-separated list of values, one per active segment.
Range: 0 Hz to 100 Hz, Unit: Hz

7.4.6.5.2 Current

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:CURRENT
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:CURRENT

```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Obw: List[float]: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: MHz
- Lower_Freq: List[enums.ResultStatus2]: float Lower frequency of the occupied bandwidth. Range: -8 MHz to 0 MHz , Unit: MHz
- Upper_Freq: List[enums.ResultStatus2]: float Upper frequency of the occupied bandwidth. Range: 0 MHz to 8 MHz , Unit: MHz

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Obw: List[float]: float Occupied bandwidth Range: 0 MHz to 8 MHz , Unit: MHz
- Lower_Freq: List[float]: float Lower frequency of the occupied bandwidth. Range: -8 MHz to 0 MHz , Unit: MHz
- Upper_Freq: List[float]: float Upper frequency of the occupied bandwidth. Range: 0 MHz to 8 MHz , Unit: MHz

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:CURRent
value: CalculateStruct = driver.multiEval.listPy.obw.current.calculate()
```

Returns CURRent occupied bandwidth (OBW) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:CURRent
value: FetchStruct = driver.multiEval.listPy.obw.current.fetch()
```

Returns CURRent occupied bandwidth (OBW) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.5.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Obw: List[float]: float Occupied bandwidth Range: 0 MHz to 8MHz (SDEVIation 0 MHz to 4 MHz), Unit: Hz

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Obw: List[float]: float Occupied bandwidth Range: 0 MHz to 8MHz (SDEviation 0 MHz to 4 MHz), Unit: Hz

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:AVERage
value: CalculateStruct = driver.multiEval.listPy.obw.average.calculate()
```

Returns occupied bandwidth (OBW) results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:OBW:AVERage
value: FetchStruct = driver.multiEval.listPy.obw.average.fetch()
```

Returns occupied bandwidth (OBW) results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.5.4 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Obw: List[float]: float Occupied bandwidth Range: 0 MHz to 8MHz (SDEviation 0 MHz to 4 MHz), Unit: Hz

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Obw: List[float]: float Occupied bandwidth Range: 0 MHz to 8MHz (SDEviation 0 MHz to 4 MHz), Unit: Hz

calculate() → CalculateStruct

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:MAXimum
value: CalculateStruct = driver.multiEval.listPy.obw.maximum.calculate()

```

Returns occupied bandwidth (OBW) results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {.. }seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:MAXimum
value: FetchStruct = driver.multiEval.listPy.obw.maximum.fetch()
```

Returns occupied bandwidth (OBW) results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.5.5 StandardDev

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:SDEviation
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:SDEviation
```

class StandardDev

StandardDev commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Obw: List[float]: float Occupied bandwidth Range: 0 MHz to 8MHz (SDEviation 0 MHz to 4 MHz), Unit: Hz

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Obw: List[float]: float Occupied bandwidth Range: 0 MHz to 8MHz (SDEviation 0 MHz to 4 MHz), Unit: Hz

calculate() → CalculateStruct


```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:SDEviation
value: CalculateStruct = driver.multiEval.listPy.obw.standardDev.calculate()
```

Returns occupied bandwidth (OBW) results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:OBW:SDEviation
value: FetchStruct = driver.multiEval.listPy.obw.standardDev.fetch()
```

Returns occupied bandwidth (OBW) results in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.value) . To define the statistical length for AVERage, MAXimum and SDEviation calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Spectrum.set. The ranges indicated below apply to all results except standard deviation results. The minimum for standard deviation results equals 0. The maximum equals the width of the indicated range divided by two. Exceptions are explicitly stated. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.6 Cp

class Cp

Cp commands group definition. 72 total commands, 11 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cp.clone()
```

Subgroups

7.4.6.6.1 RpiCh

class RpiCh

RpiCh commands group definition. 9 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cp.rpiCh.clone()
```

Subgroups

7.4.6.6.1.1 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RPiCh:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RPiCh:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cp.rpiCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states, one per active segment.

7.4.6.6.1.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RPiCh:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RPiCh:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:CURRent
value: List[float] = driver.multiEval.listPy.cp.rpiCh.current.calculate()
```

Returns the RMS power (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:CURRent
value: List[float] = driver.multiEval.listPy.cp.rpiCh.current.fetch()
```

Returns the RMS power (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.1.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:AVERage
value: List[float] = driver.multiEval.listPy.cp.rpiCh.average.calculate()
```

Returns the RMS power (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:AVERage
value: List[float] = driver.multiEval.listPy.cp.rpiCh.average.fetch()
```

Returns the RMS power (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.1.4 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rpiCh.maximum.calculate()
```

Returns the RMS power (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rpiCh.maximum.fetch()
```

Returns the RMS power (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.1.5 Minimum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:MINimum
value: List[float] = driver.multiEval.listPy.cp.rpiCh.minimum.calculate()
```

Returns the RMS power (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RPICh:MINimum
value: List[float] = driver.multiEval.listPy.cp.rpiCh.minimum.fetch()
```

Returns the RMS power (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.2 RdcCh

class RdcCh

RdcCh commands group definition. 9 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cp.rdcCh.clone()
```

Subgroups

7.4.6.6.2.1 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cp.rdcCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states, one per active segment.

7.4.6.6.2.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:CURREnt
value: List[float] = driver.multiEval.listPy.cp.rdcCh.current.calculate()
```

Returns the RMS power (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:CURREnt
value: List[float] = driver.multiEval.listPy.cp.rdcCh.current.fetch()
```

Returns the RMS power (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.2.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:AVERage
value: List[float] = driver.multiEval.listPy.cp.rdcCh.average.calculate()
```

Returns the RMS power (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:AVERage
value: List[float] = driver.multiEval.listPy.cp.rdcCh.average.fetch()
```

Returns the RMS power (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.2.4 Maximum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rdcCh.maximum.calculate()

```

Returns the RMS power (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rdcCh.maximum.fetch()

```

Returns the RMS power (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.2.5 Minimum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:MINimum

```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:MINimum
value: List[float] = driver.multiEval.listPy.cp.rdcCh.minimum.calculate()

```


Returns the RMS power (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RDCCh:MINimum
value: List[float] = driver.multiEval.listPy.cp.rdcCh.minimum.fetch()
```

Returns the RMS power (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.3 RccCh

class RccCh

RccCh commands group definition. 9 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cp.rccCh.clone()
```

Subgroups

7.4.6.6.3.1 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cp.rccCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states,
one per active segment.

7.4.6.6.3.2 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:CURRent
value: List[float] = driver.multiEval.listPy.cp.rccCh.current.calculate()
```

Returns the RMS power (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:CURRent
value: List[float] = driver.multiEval.listPy.cp.rccCh.current.fetch()
```

Returns the RMS power (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

7.4.6.6.3.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:AVERage
value: List[float] = driver.multiEval.listPy.cp.rccCh.average.calculate()
```

Returns the RMS power (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:AVERage
value: List[float] = driver.multiEval.listPy.cp.rccCh.average.fetch()
```

Returns the RMS power (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

7.4.6.6.3.4 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rccCh.maximum.calculate()
```

Returns the RMS power (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rccCh.maximum.fetch()
```

Returns the RMS power (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

7.4.6.6.3.5 Minimum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:MINimum
value: List[float] = driver.multiEval.listPy.cp.rccCh.minimum.calculate()
```

Returns the RMS power (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RCCCh:MINimum
value: List[float] = driver.multiEval.listPy.cp.rccCh.minimum.fetch()
```

Returns the RMS power (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

7.4.6.6.4 ReaCh

class ReaCh

ReaCh commands group definition. 9 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cp.reaCh.clone()
```

Subgroups

7.4.6.6.4.1 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cp.reaCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states, one per active segment.

7.4.6.6.4.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:CURREnt
value: List[float] = driver.multiEval.listPy.cp.reaCh.current.calculate()
```

Returns the RMS power (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:REACH:CURRent
value: List[float] = driver.multiEval.listPy.cp.reaCh.current.fetch()
```

Returns the RMS power (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.4.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:REACH:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:REACH:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:REACH:AVERage
value: List[float] = driver.multiEval.listPy.cp.reaCh.average.calculate()
```

Returns the RMS power (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:REACH:AVERage
value: List[float] = driver.multiEval.listPy.cp.reaCh.average.fetch()
```

Returns the RMS power (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.4.4 Maximum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:MAXimum
value: List[float] = driver.multiEval.listPy.cp.reaCh.maximum.calculate()
```

Returns the RMS power (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:MAXimum
value: List[float] = driver.multiEval.listPy.cp.reaCh.maximum.fetch()
```

Returns the RMS power (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.4.5 Minimum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:REACH:MINimum
value: List[float] = driver.multiEval.listPy.cp.reaCh.minimum.calculate()
```

Returns the RMS power (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:REACH:MINimum
value: List[float] = driver.multiEval.listPy.cp.reaCh.minimum.fetch()
```

Returns the RMS power (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-60 dB to 0 dB , Unit: dB

7.4.6.6.5 Rfch

class Rfch

Rfch commands group definition. 9 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cp.rfch.clone()
```

Subgroups

7.4.6.6.5.1 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]


```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cp.rfch.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: INVisible | ACTive | IACTive | ALIased Comma-separated list of states, one per active segment.

7.4.6.6.5.2 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:CURREnt
value: List[float] = driver.multiEval.listPy.cp.rfch.current.calculate()
```

Returns the RMS power (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:CURREnt
value: List[float] = driver.multiEval.listPy.cp.rfch.current.fetch()
```

Returns the RMS power (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.5.3 Average

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:AVERage
value: List[float] = driver.multiEval.listPy.cp.rfch.average.calculate()

```

Returns the RMS power (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:AVERage
value: List[float] = driver.multiEval.listPy.cp.rfch.average.fetch()

```

Returns the RMS power (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.5.4 Maximum

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rfch.maximum.calculate()

```

Returns the RMS power (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rfch.maximum.fetch()
```

Returns the RMS power (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.5.5 Minimum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:MINimum
value: List[float] = driver.multiEval.listPy.cp.rfch.minimum.calculate()
```

Returns the RMS power (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RFCH:MINimum
value: List[float] = driver.multiEval.listPy.cp.rfch.minimum.fetch()
```

Returns the RMS power (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.6 RsCh

class RsCh

RsCh commands group definition. 18 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cp.rsCh.clone()
```

Subgroups

7.4.6.6.6.1 Ztef

class Ztef

Ztef commands group definition. 9 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cp.rsCh.ztef.clone()
```

Subgroups

7.4.6.6.6.2 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:ZTEF:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:ZTEF:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cp.rsCh.ztef.state.
↪ fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_02_e_04: INVisible | ACTive | IACTive | ALIased Comma-separated list of states, one per active segment.

7.4.6.6.3 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZTEF:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZTEF:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEValuation:LIST:CP:RSCH:ZTEF:CURRent
value: List[float] = driver.multiEval.listPy.cp.rsCh.ztef.current.calculate()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: 'Zero-Two-Exponent-Four'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZTEF:CURRent
value: List[float] = driver.multiEval.listPy.cp.rsCh.ztef.current.fetch()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: 'Zero-Two-Exponent-Four'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.4 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZTEF:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZTEF:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
→ :MEValuation:LIST:CP:RSCH:ZTEF:AVERage
value: List[float] = driver.multiEval.listPy.cp.rsCh.ztef.average.calculate()

```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZTEF:AVERage
value: List[float] = driver.multiEval.listPy.cp.rsCh.ztef.average.fetch()

```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.5 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZTEF:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZTEF:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:CP:RSCH:ZTEF:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rsCh.ztef.maximum.calculate()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:ZTEF:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rsCh.ztef.maximum.fetch()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.6 Minimum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:ZTEF:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:ZTEF:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:CP:RSCH:ZTEF:MINimum
value: List[float] = driver.multiEval.listPy.cp.rsCh.ztef.minimum.calculate()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZTEF:MINimum
value: List[float] = driver.multiEval.listPy.cp.rsCh.ztef.minimum.fetch()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.6.7 Zoet

class Zoet

Zoet commands group definition. 9 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cp.rsCh.zoet.clone()
```

Subgroups

7.4.6.6.6.8 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cp.rsCh.zoet.state.
↪ fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: INVisible | ACTIVE | IACTIVE | ALIased Comma-separated
list of states, one per active segment.

7.4.6.6.9 Current

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:CURRent

```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>
→:MEValuation:LIST:CP:RSCH:ZOET:CURRent
value: List[float] = driver.multiEval.listPy.cp.rsCh.zoet.current.calculate()

```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:CURRent
value: List[float] = driver.multiEval.listPy.cp.rsCh.zoet.current.fetch()

```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.10 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:AVERAge
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:AVERAge

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:CP:RSCH:ZOET:AVERage
value: List[float] = driver.multiEval.listPy.cp.rsCh.zoet.average.calculate()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:ZOET:AVERage
value: List[float] = driver.multiEval.listPy.cp.rsCh.zoet.average.fetch()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.11 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:ZOET:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:ZOET:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:CP:RSCH:ZOET:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rsCh.zoet.maximum.calculate()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:MAXimum
value: List[float] = driver.multiEval.listPy.cp.rsCh.zoet.maximum.fetch()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.12 Minimum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:CP:RSCH:ZOET:MINimum
value: List[float] = driver.multiEval.listPy.cp.rsCh.zoet.minimum.calculate()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:RSCH:ZOET:MINimum
value: List[float] = driver.multiEval.listPy.cp.rsCh.zoet.minimum.fetch()
```

Returns the RMS power (statistical values) of reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -60 dB to 0 dB , Unit: dB

7.4.6.6.7 Current

SCPI Commands

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:CURRent

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rdc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rcc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rea_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rfch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_02_E_04: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_01_E_02: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.

- Rpi_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rdc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rcc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rea_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rfch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_02_E_04: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_01_E_02: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:CURRENT
value: CalculateStruct = driver.multiEval.listPy.cp.current.calculate()
```

Returns channel power (CP) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:CURRENT
value: FetchStruct = driver.multiEval.listPy.cp.current.fetch()
```

Returns channel power (CP) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.6.8 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rdc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rcc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rea_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rfch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_02_E_04: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_01_E_02: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rdc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB

- **Rcc_Ch**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rea_Ch**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rfch**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rsch_0_W_02_E_04**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rsch_0_W_01_E_02**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rsch_1_W_06_E_08**: List[float]: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:AVERage
value: CalculateStruct = driver.multiEval.listPy.cp.average.calculate()
```

Returns channel power (CP) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:AVERage
value: FetchStruct = driver.multiEval.listPy.cp.average.fetch()
```

Returns channel power (CP) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERage, MAXimum, MINimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.6.9 Maximum

SCPI Commands

```
FEtCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rdc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rcc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rea_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rfch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_02_E_04: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_01_E_02: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rdc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB

- **Rcc_Ch**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rea_Ch**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rfch**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rsch_0_W_02_E_04**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rsch_0_W_01_E_02**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rsch_1_W_06_E_08**: List[float]: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:MAXimum
value: CalculateStruct = driver.multiEval.listPy.cp.maximum.calculate()
```

Returns channel power (CP) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:MAXimum
value: FetchStruct = driver.multiEval.listPy.cp.maximum.fetch()
```

Returns channel power (CP) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets { } are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.6.10 Minimum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:MINimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rdc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rcc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rea_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rfch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_02_E_04: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_0_W_01_E_02: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- Rdc_Ch: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB

- **Rcc_Ch**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rea_Ch**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rfch**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rsch_0_W_02_E_04**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rsch_0_W_01_E_02**: List[float]: float RMS channel power values for the indicated channels. Range: -60 dB to 0 dB (SDEViation 0 dB to 60 dB) , Unit: dB
- **Rsch_1_W_06_E_08**: List[float]: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:MINimum
value: CalculateStruct = driver.multiEval.listPy.cp.minimum.calculate()
```

Returns channel power (CP) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:MINimum
value: FetchStruct = driver.multiEval.listPy.cp.minimum.fetch()
```

Returns channel power (CP) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.value) . To define the statistical length for AVERAGE, MAXimum, MINimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval. ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.6.11 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: decimal The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated. Range: 0 | 3 | 4 | 8
- Rpi_Ch: List[enums.SigChStateA]: INVIsible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rdc_Ch: List[enums.SigChStateA]: INVIsible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rcc_Ch: List[enums.SigChStateA]: INVIsible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rea_Ch: List[enums.SigChStateA]: INVIsible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rfch: List[enums.SigChStateA]: INVIsible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rsch_0_W_02_E_04: List[enums.SigChStateA]: INVIsible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rsch_0_W_01_E_02: List[enums.SigChStateA]: INVIsible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rsch_1_W_06_E_08: List[enums.SigChStateA]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[enums.SigChStateA]: float For future use - returned value not relevant.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:STATe
value: FetchStruct = driver.multiEval.listPy.cp.state.fetch()
```

Return the states of the channels for power measurement (CP) . The values listed below in curly brackets { } are returned for each active segment: { ... }seg 1, { ... }seg 2, ..., { ... }seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.7 Cpo

class Cpo

Cpo commands group definition. 56 total commands, 10 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cpo.clone()
```

Subgroups

7.4.6.7.1 RpiCh

class RpiCh

RpiCh commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cpo.rpiCh.clone()
```

Subgroups

7.4.6.7.1.1 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RPICh:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RPICh:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cpo.rpiCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states, one per active segment.

7.4.6.7.1.2 Current

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICH:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICH:CURRent

```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICH:CURRent
value: List[float] = driver.multiEval.listPy.cpo.rpiCh.current.calculate()

```

Returns the phase offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

fetch() → List[float]

```

# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICH:CURRent
value: List[float] = driver.multiEval.listPy.cpo.rpiCh.current.fetch()

```

Returns the phase offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

7.4.6.7.1.3 Average

SCPI Commands

```

FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICH:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICH:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICH:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rpiCh.average.calculate()

```

Returns the phase offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICh:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rpiCh.average.fetch()
```

Returns the phase offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

7.4.6.7.1.4 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICh:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICh:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICh:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rpiCh.maximum.calculate()
```

Returns the phase offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RPICh:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rpiCh.maximum.fetch()
```

Returns the phase offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.2 RdcCh

class RdcCh

RdcCh commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cpo.rdcCh.clone()
```

Subgroups

7.4.6.7.2.1 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RDCCh:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RDCCh:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cpo.rdcCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: INVisible | ACTive | IACTive | ALIased Comma-separated list of states,
one per active segment.

7.4.6.7.2.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RDCCh:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RDCCh:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:CURRent
value: List[float] = driver.multiEval.listPy.cpo.rdcCh.current.calculate()
```

Returns the phase offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:CURRent
value: List[float] = driver.multiEval.listPy.cpo.rdcCh.current.fetch()
```

Returns the phase offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.2.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rdcCh.average.calculate()
```

Returns the phase offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rdcCh.average.fetch()
```

Returns the phase offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.2.4 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rdcCh.maximum.calculate()
```

Returns the phase offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RDCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rdcCh.maximum.fetch()
```

Returns the phase offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.3 RccCh

class RccCh

RccCh commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cpo.rccCh.clone()
```

Subgroups

7.4.6.7.3.1 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RCCCh:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RCCCh:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cpo.rccCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states, one per active segment.

7.4.6.7.3.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RCCCh:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RCCCh:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RCCCh:CURREnt
value: List[float] = driver.multiEval.listPy.cpo.rccCh.current.calculate()
```

Returns the phase offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RCCCh:CURRent
value: List[float] = driver.multiEval.listPy.cpo.rccCh.current.fetch()
```

Returns the phase offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.3.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RCCCh:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RCCCh:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RCCCh:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rccCh.average.calculate()
```

Returns the phase offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RCCCh:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rccCh.average.fetch()
```

Returns the phase offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.3.4 Maximum

SCPI Commands

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RCCCh:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RCCCh:MAXimum

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RCCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rccCh.maximum.calculate()
```

Returns the phase offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RCCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rccCh.maximum.fetch()
```

Returns the phase offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.4 ReaCh

class ReaCh

ReaCh commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cpo.reaCh.clone()
```

Subgroups

7.4.6.7.4.1 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:REACH:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:REACH:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cpo.reaCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states, one per active segment.

7.4.6.7.4.2 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:REACH:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:REACH:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:REACH:CURREnt
value: List[float] = driver.multiEval.listPy.cpo.reaCh.current.calculate()
```

Returns the phase offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:REACH:CURRENT
value: List[float] = driver.multiEval.listPy.cpo.reaCh.current.fetch()
```

Returns the phase offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.4.3 Average

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:REACH:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:REACH:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:REACH:AVERage
value: List[float] = driver.multiEval.listPy.cpo.reaCh.average.calculate()
```

Returns the phase offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:REACH:AVERage
value: List[float] = driver.multiEval.listPy.cpo.reaCh.average.fetch()
```

Returns the phase offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.4.4 Maximum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:REACH:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:REACH:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:REACH:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.reaCh.maximum.calculate()
```

Returns the phase offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:REACH:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.reaCh.maximum.fetch()
```

Returns the phase offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-180 deg to +180 deg

7.4.6.7.5 Rfch

class Rfch

Rfch commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cpo.rfch.clone()
```


Subgroups

7.4.6.7.5.1 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RFCH:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RFCH:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cpo.rfch.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: INVisible | ACTive | IACTive | ALIased Comma-separated list of states, one per active segment.

7.4.6.7.5.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RFCH:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RFCH:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RFCH:CURREnt
value: List[float] = driver.multiEval.listPy.cpo.rfch.current.calculate()
```

Returns the phase offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RFCH:CURREnt
value: List[float] = driver.multiEval.listPy.cpo.rfch.current.fetch()
```

Returns the phase offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

7.4.6.7.5.3 Average

SCPI Commands

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RFCH:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RFCH:AVERage

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RFCH:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rfch.average.calculate()
```

Returns the phase offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RFCH:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rfch.average.fetch()
```

Returns the phase offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

7.4.6.7.5.4 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RFCH:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RFCH:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RFCH:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rfch.maximum.calculate()

```

Returns the phase offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RFCH:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rfch.maximum.fetch()

```

Returns the phase offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

7.4.6.7.6 RsCh

class RsCh

RsCh commands group definition. 14 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```

# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cpo.rsCh.clone()

```

Subgroups

7.4.6.7.6.1 Ztef

class Ztef

Ztef commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cpo.rsCh.ztef.clone()
```

Subgroups

7.4.6.7.6.2 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZTEF:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZTEF:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cpo.rsCh.ztef.state.
↳ fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_02_e_04: INVisible | ACTive | IACTive | ALIased Comma-separated list of states, one per active segment.

7.4.6.7.6.3 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZTEF:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZTEF:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:CPO:RSCH:ZTEF:CURRent
value: List[float] = driver.multiEval.listPy.cpo.rsCh.ztef.current.calculate()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZTEF:CURRent
value: List[float] = driver.multiEval.listPy.cpo.rsCh.ztef.current.fetch()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -180 deg to +180 deg

7.4.6.7.6.4 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZTEF:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZTEF:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:CPO:RSCH:ZTEF:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rsCh.ztef.average.calculate()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RSCH:ZTEF:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rsCh.ztef.average.fetch()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -180 deg to +180 deg

7.4.6.7.6.5 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RSCH:ZTEF:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RSCH:ZTEF:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:CPO:RSCH:ZTEF:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rsCh.ztef.maximum.calculate()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RSCH:ZTEF:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rsCh.ztef.maximum.fetch()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -180 deg to +180 deg

7.4.6.7.6.6 Zoet

class Zoet

Zoet commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cpo.rsCh.zoet.clone()
```

Subgroups

7.4.6.7.6.7 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZOET:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZOET:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cpo.rsCh.zoet.state.
↪ fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: INVisible | ACTive | IACTive | ALIased Comma-separated list of states, one per active segment.

7.4.6.7.6.8 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZOET:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZOET:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:CPO:RSCH:ZOET:CURRENT
value: List[float] = driver.multiEval.listPy.cpo.rsCh.zoet.current.calculate()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: ‘Zero-One-Exponent-Two’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZOET:CURRENT
value: List[float] = driver.multiEval.listPy.cpo.rsCh.zoet.current.fetch()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: ‘Zero-One-Exponent-Two’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

7.4.6.7.6.9 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZOET:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:RSCH:ZOET:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:CPO:RSCH:ZOET:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rsCh.zoet.average.calculate()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: ‘Zero-One-Exponent-Two’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RSCH:ZOET:AVERage
value: List[float] = driver.multiEval.listPy.cpo.rsCh.zoet.average.fetch()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

7.4.6.7.6.10 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RSCH:ZOET:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RSCH:ZOET:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:CPO:RSCH:ZOET:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rsCh.zoet.maximum.calculate()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:RSCH:ZOET:MAXimum
value: List[float] = driver.multiEval.listPy.cpo.rsCh.zoet.maximum.fetch()
```

Returns the phase offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: 'Zero-One-Exponent-Two'. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -180 deg to +180 deg

7.4.6.7.7 Current

SCPI Commands

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:CURRENT
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:CURRENT

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rfch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_02_E_04: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_01_E_02: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.

- Rpi_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rfch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_02_E_04: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_01_E_02: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:CURRENT
value: CalculateStruct = driver.multiEval.listPy.cpo.current.calculate()
```

Returns channel phase offset (CPO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:CURRENT
value: FetchStruct = driver.multiEval.listPy.cpo.current.fetch()
```

Returns channel phase offset (CPO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.7.8 Average

SCPI Commands

<code>FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CP0:AVERage</code> <code>CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CP0:AVERage</code>
--

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rfch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_02_E_04: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_01_E_02: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg

- **Rcc_Ch**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rea_Ch**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rfch**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rsch_0_W_02_E_04**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rsch_0_W_01_E_02**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rsch_1_W_06_E_08**: List[float]: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:AVERage
value: CalculateStruct = driver.multiEval.listPy.cpo.average.calculate()
```

Returns channel phase offset (CPO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:AVERage
value: FetchStruct = driver.multiEval.listPy.cpo.average.fetch()
```

Returns channel phase offset (CPO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.7.9 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CPO:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rcc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rea_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rfch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_02_E_04: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_0_W_01_E_02: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- Rdc_Ch: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg

- **Rcc_Ch**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rea_Ch**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rfch**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rsch_0_W_02_E_04**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rsch_0_W_01_E_02**: List[float]: float Phase offset for the indicated channels in the in-phase signal path (I-signal) . Range: -180 deg to +180 deg Unit: deg
- **Rsch_1_W_06_E_08**: List[float]: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:MAXimum
value: CalculateStruct = driver.multiEval.listPy.cpo.maximum.calculate()
```

Returns channel phase offset (CPO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CPO:MAXimum
value: FetchStruct = driver.multiEval.listPy.cpo.maximum.fetch()
```

Returns channel phase offset (CPO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.7.10 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rdc_Ch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rcc_Ch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rea_Ch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rfch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rsch_0_W_02_E_04: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rsch_0_W_01_E_02: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rsch_1_W_06_E_08: List[enums.SigChStateA]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[enums.SigChStateA]: float For future use - returned value not relevant.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:STATE
value: FetchStruct = driver.multiEval.listPy.cpo.state.fetch()
```

Return the states of the channels for channel phase offset (CPO) measurements. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.8 Cto

class Cto

Cto commands group definition. 56 total commands, 10 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cto.clone()
```

Subgroups

7.4.6.8.1 RpiCh

class RpiCh

RpiCh commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cto.rpiCh.clone()
```

Subgroups

7.4.6.8.1.1 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RPICH:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RPICH:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cto.rpiCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states, one per active segment.

7.4.6.8.1.2 Current

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:CURRent

```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:CURRent
value: List[float] = driver.multiEval.listPy.cto.rpiCh.current.calculate()

```

Returns the time offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:CURRent
value: List[float] = driver.multiEval.listPy.cto.rpiCh.current.fetch()

```

Returns the time offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.1.3 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:AVERage
value: List[float] = driver.multiEval.listPy.cto.rpiCh.average.calculate()

```

Returns the time offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:AVERage
value: List[float] = driver.multiEval.listPy.cto.rpiCh.average.fetch()
```

Returns the time offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.1.4 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rpiCh.maximum.calculate()
```

Returns the time offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RPICh:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rpiCh.maximum.fetch()
```

Returns the time offset (statistical values) of the reverse pilot channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rpi_ch: float Comma-separated list of values, one per active segment. Range:
-50 ns to 50 ns, Unit: s

7.4.6.8.2 RdcCh

class RdcCh

RdcCh commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cto.rdcCh.clone()
```

Subgroups

7.4.6.8.2.1 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RDCCh:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RDCCh:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cto.rdcCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states,
one per active segment.

7.4.6.8.2.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RDCCh:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RDCCh:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:CURRent
value: List[float] = driver.multiEval.listPy.cto.rdcCh.current.calculate()
```

Returns the time offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:CURRent
value: List[float] = driver.multiEval.listPy.cto.rdcCh.current.fetch()
```

Returns the time offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Range: -50 ns to 50 ns, Unit: s

7.4.6.8.2.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:AVERage
value: List[float] = driver.multiEval.listPy.cto.rdcCh.average.calculate()
```

Returns the time offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:AVERage
value: List[float] = driver.multiEval.listPy.cto.rdcCh.average.fetch()
```

Returns the time offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Range: -50 ns to 50 ns, Unit: s

7.4.6.8.2.4 Maximum

SCPI Commands

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:MAXimum

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rdcCh.maximum.calculate()
```

Returns the time offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RDCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rdcCh.maximum.fetch()
```

Returns the time offset (statistical values) of the reverse dedicated control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rdc_ch: float Range: -50 ns to 50 ns, Unit: s

7.4.6.8.3 RccCh

class RccCh

RccCh commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cto.rccCh.clone()
```

Subgroups

7.4.6.8.3.1 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cto.rccCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states, one per active segment.

7.4.6.8.3.2 Current

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:CURRent
value: List[float] = driver.multiEval.listPy.cto.rccCh.current.calculate()
```

Returns the time offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:CURRent
value: List[float] = driver.multiEval.listPy.cto.rccCh.current.fetch()
```

Returns the time offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.3.3 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:AVERage
value: List[float] = driver.multiEval.listPy.cto.rccCh.average.calculate()
```

Returns the time offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:AVERage
value: List[float] = driver.multiEval.listPy.cto.rccCh.average.fetch()
```

Returns the time offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.3.4 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rccCh.maximum.calculate()

```

Returns the time offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RCCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rccCh.maximum.fetch()

```

Returns the time offset (statistical values) of the reverse common control channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rcc_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.4 ReaCh

class ReaCh

ReaCh commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```

# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cto.reaCh.clone()

```

Subgroups

7.4.6.8.4.1 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cto.reaCh.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: INVisible | ACTive | IACTive | ALLased Comma-separated list of states, one per active segment.

7.4.6.8.4.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:CURRent
value: List[float] = driver.multiEval.listPy.cto.reaCh.current.calculate()
```

Returns the time offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:CURRent
value: List[float] = driver.multiEval.listPy.cto.reaCh.current.fetch()
```

Returns the time offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.4.3 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:REACH:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:REACH:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:REACH:AVERage
value: List[float] = driver.multiEval.listPy.cto.reaCh.average.calculate()

```

Returns the time offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:REACH:AVERage
value: List[float] = driver.multiEval.listPy.cto.reaCh.average.fetch()

```

Returns the time offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.4.4 Maximum

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:MAXimum
value: List[float] = driver.multiEval.listPy.cto.reaCh.maximum.calculate()
```

Returns the time offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REACH:MAXimum
value: List[float] = driver.multiEval.listPy.cto.reaCh.maximum.fetch()
```

Returns the time offset (statistical values) of the reverse enhanced access channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rea_ch: float Comma-separated list of values, one per active segment. Range:
-50 ns to 50 ns, Unit: s

7.4.6.8.5 Rfch

class Rfch

Rfch commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cto.rfch.clone()
```

Subgroups

7.4.6.8.5.1 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RFCH:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RFCH:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cto.rfch.state.fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: INVisible | ACTive | IACTive | ALIased Comma-separated list of states, one per active segment.

7.4.6.8.5.2 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RFCH:CURREnt
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RFCH:CURREnt
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RFCH:CURREnt
value: List[float] = driver.multiEval.listPy.cto.rfch.current.calculate()
```

Returns the time offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RFCH:CURREnt
value: List[float] = driver.multiEval.listPy.cto.rfch.current.fetch()
```

Returns the time offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.5.3 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RFCH:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RFCH:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```

# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RFCH:AVERage
value: List[float] = driver.multiEval.listPy.cto.rfch.average.calculate()

```

Returns the time offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```

# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RFCH:AVERage
value: List[float] = driver.multiEval.listPy.cto.rfch.average.fetch()

```

Returns the time offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.5.4 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RFCH:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RFCH:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RFCH:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rfch.maximum.calculate()
```

Returns the time offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RFCH:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rfch.maximum.fetch()
```

Returns the time offset (statistical values) of the reverse fundamental channel for all active list mode segments. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rfch: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.6 RsCh

class RsCh

RsCh commands group definition. 14 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cto.rsCh.clone()
```

Subgroups

7.4.6.8.6.1 Ztef

class Ztef

Ztef commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cto.rsCh.ztef.clone()
```

Subgroups

7.4.6.8.6.2 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZTEF:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZTEF:STATE
value: List[enums.SigChStateA] = driver.multiEval.listPy.cto.rsCh.ztef.state.
↳ fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_02_e_04: INVisible | ACTive | IACTive | ALIased Comma-separated list of states, one per active segment.

7.4.6.8.6.3 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZTEF:CURRENT
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZTEF:CURRENT
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]


```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:CTO:RSCH:ZTEF:CURRent
value: List[float] = driver.multiEval.listPy.cto.rsCh.ztef.current.calculate()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZTEF:CURRent
value: List[float] = driver.multiEval.listPy.cto.rsCh.ztef.current.fetch()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -50 ns to 50 ns, Unit: s

7.4.6.8.6.4 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZTEF:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZTEF:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEvaluation:LIST:CTO:RSCH:ZTEF:AVERage
value: List[float] = driver.multiEval.listPy.cto.rsCh.ztef.average.calculate()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RSCH:ZTEF:AVERage
value: List[float] = driver.multiEval.listPy.cto.rsCh.ztef.average.fetch()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -50 ns to 50 ns, Unit: s

7.4.6.8.6.5 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RSCH:ZTEF:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RSCH:ZTEF:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:CTO:RSCH:ZTEF:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rsCh.ztef.maximum.calculate()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RSCH:ZTEF:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rsCh.ztef.maximum.fetch()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZTEF refers to the corresponding Walsh code W024: ‘Zero-Two-Exponent-Four’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rschw_02_e_04: float Comma-separated list of values, one per active segment.
Range: -50 ns to 50 ns, Unit: s

7.4.6.8.6.6 Zoet

class Zoet

Zoet commands group definition. 7 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cto.rsCh.zoet.clone()
```

Subgroups

7.4.6.8.6.7 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZOET:STATe
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[RsCmwCdma2kMeas.enums.SigChStateA]

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZOET:STATe
value: List[enums.SigChStateA] = driver.multiEval.listPy.cto.rsCh.zoet.state.
↪ fetch()
```

Returns the state of a particular reverse link channel (R-CCCH, R-DCCH, R-EACH, R-FCH, R-PICH, R-SCH0 - W0102, R-SCH0 - W0204) in a channel-related measurement (CP, CPO, CTO) for all active segments.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: INVisible | ACTive | IACTive | ALIased Comma-separated list of states, one per active segment.

7.4.6.8.6.8 Current

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZOET:CURRent
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZOET:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:CTO:RSCH:ZOET:CURRENT
value: List[float] = driver.multiEval.listPy.cto.rsCh.zoet.current.calculate()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: ‘Zero-One-Exponent-Two’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZOET:CURRENT
value: List[float] = driver.multiEval.listPy.cto.rsCh.zoet.current.fetch()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: ‘Zero-One-Exponent-Two’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.6.9 Average

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZOET:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RSCH:ZOET:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↳:MEvaluation:LIST:CTO:RSCH:ZOET:AVERage
value: List[float] = driver.multiEval.listPy.cto.rsCh.zoet.average.calculate()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: ‘Zero-One-Exponent-Two’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RSCH:ZOET:AVERage
value: List[float] = driver.multiEval.listPy.cto.rsCh.zoet.average.fetch()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: ‘Zero-One-Exponent-Two’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.6.10 Maximum

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RSCH:ZOET:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RSCH:ZOET:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

calculate() → List[float]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>
↪:MEValuation:LIST:CTO:RSCH:ZOET:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rsCh.zoet.maximum.calculate()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: ‘Zero-One-Exponent-Two’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

fetch() → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:RSCH:ZOET:MAXimum
value: List[float] = driver.multiEval.listPy.cto.rsCh.zoet.maximum.fetch()
```

Returns the time offset (statistical values) of the reverse supplemental channel 0 for all active list mode segments. The mnemonic ZOET refers to the corresponding Walsh code W012: ‘Zero-One-Exponent-Two’. The values described below are returned by FETCh commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

return rsch_0_w_01_e_02: float Comma-separated list of values, one per active segment. Range: -50 ns to 50 ns, Unit: s

7.4.6.8.7 Current

SCPI Commands

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CT0:CURRENT
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CT0:CURRENT

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rdc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rcc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rea_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rfch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_02_E_04: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_01_E_02: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.

- Rpi_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rdc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rcc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rea_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rfch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_02_E_04: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_01_E_02: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:CURRENT
value: CalculateStruct = driver.multiEval.listPy.cto.current.calculate()
```

Returns channel time offset (CTO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:CURRENT
value: FetchStruct = driver.multiEval.listPy.cto.current.fetch()
```

Returns channel time offset (CTO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.8.8 Average

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CT0:AVERage
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CT0:AVERage

```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rdc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rcc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rea_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rfch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_02_E_04: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_01_E_02: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rdc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s

- **Rcc_Ch**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rea_Ch**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rfch**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_0_W_02_E_04**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_0_W_01_E_02**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_1_W_06_E_08**: List[float]: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:AVERage
value: CalculateStruct = driver.multiEval.listPy.cto.average.calculate()
```

Returns channel time offset (CTO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:AVERage
value: FetchStruct = driver.multiEval.listPy.cto.average.fetch()
```

Returns channel time offset (CTO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.8.9 Maximum

SCPI Commands

```

FETCh:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:MAXimum
CALCulate:CDMA:MEASurement<Instance>:MEValuation:LIST:CTO:MAXimum

```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rdc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rcc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rea_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rfch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_02_E_04: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_0_W_01_E_02: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rsch_1_W_06_E_08: List[float]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[float]: float For future use - returned value not relevant.

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- Rdc_Ch: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s

- **Rcc_Ch**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rea_Ch**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rfch**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_0_W_02_E_04**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_0_W_01_E_02**: List[float]: float Time offset for the indicated channels in the in-phase signal path (I-signal) . Range: -50 ns to +50 ns Unit: s
- **Rsch_1_W_06_E_08**: List[float]: float For future use - returned value not relevant.
- **Rsch_1_W_02_E_04**: List[float]: float For future use - returned value not relevant.

calculate() → CalculateStruct

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:MAXimum
value: CalculateStruct = driver.multiEval.listPy.cto.maximum.calculate()
```

Returns channel time offset (CTO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:MAXimum
value: FetchStruct = driver.multiEval.listPy.cto.maximum.fetch()
```

Returns channel time offset (CTO) results for all active segments in list mode (see method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.value) . To define the statistical length for AVERAGE and MAXimum calculation and to enable the calculation of the results, use the command method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.Segment.Modulation.set. The values described below are returned by FETCH commands. CALCulate commands return limit check results instead, one value for each result listed below. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure. MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.6.8.10 State

SCPI Commands

```
FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:STATE
```

class State

State commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'. In list mode, a zero reliability indicator indicates that the results in all measured segments are valid. A non-zero value indicates that an error occurred in at least one of the measured segments.
- Seg_Reliability: List[int]: 0 | 3 | 4 | 8 The segment reliability indicates whether one of the following exceptions occurred in this segment: 0: No error 3: Signal overflow 4: Signal low 8: Synchronization error If a combination of exceptions occurs, the most severe error is indicated.
- Rpi_Ch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rdc_Ch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rcc_Ch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rea_Ch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rfch: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rsch_0_W_02_E_04: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rsch_0_W_01_E_02: List[enums.SigChStateA]: INVisible | ACTive | IACTive INV: No channel available ACTive: Active channel IACTive: Inactive channel ALIased: Aliased channel
- Rsch_1_W_06_E_08: List[enums.SigChStateA]: float For future use - returned value not relevant.
- Rsch_1_W_02_E_04: List[enums.SigChStateA]: float For future use - returned value not relevant.

fetch() → FetchStruct

```
# SCPI: FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:STATE
value: FetchStruct = driver.multiEval.listPy.cto.state.fetch()
```

Return the states of the channels for channel time offset (CTO) measurements. The values listed below in curly brackets {} are returned for each active segment: {...}seg 1, {...}seg 2, ..., {...}seg n. The number of active segments n is determined by method RsCmwCdma2kMeas.Configure.MultiEval.ListPy.count. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

return structure: for return value, see the help for FetchStruct structure arguments.

7.5 Oltr

SCPI Commands

```
INITiate:CDMA:MEASurement<Instance>:OLTR
ABORt:CDMA:MEASurement<Instance>:OLTR
STOP:CDMA:MEASurement<Instance>:OLTR
```

class Oltr

Oltr commands group definition. 15 total commands, 2 Sub-groups, 3 group commands

abort() → None

```
# SCPI: ABORt:CDMA:MEASurement<Instance>:OLTR
driver.oltr.abort()

INTRO_CMD_HELP: Starts, stops, or aborts the measurement:

- INITiate... starts or restarts the measurement. The measurement enters
↳ the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY
↳ ' state. Measurement results are kept. The resources remain allocated to the
↳ measurement.
- ABORt... halts the measurement immediately. The measurement enters the
↳ 'OFF' state. All measurement values are set to NAV. Allocated resources are
↳ released.
```

Use FETCh...STATe? to query the current measurement state.

abort_with_opc() → None

```
# SCPI: ABORt:CDMA:MEASurement<Instance>:OLTR
driver.oltr.abort_with_opc()

INTRO_CMD_HELP: Starts, stops, or aborts the measurement:

- INITiate... starts or restarts the measurement. The measurement enters
↳ the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY
↳ ' state. Measurement results are kept. The resources remain allocated to the
↳ measurement.
- ABORt... halts the measurement immediately. The measurement enters the
↳ 'OFF' state. All measurement values are set to NAV. Allocated resources are
↳ released.
```

Use FETCh...STATe? to query the current measurement state.

Same as abort, but waits for the operation to complete before continuing further. Use the RsCmwCdma2kMeas.utilities.opc_timeout_set() to set the timeout value.

initiate() → None

```
# SCPI: INITiate:CDMA:MEASurement<Instance>:OLTR
driver.oltr.initiate()

INTRO_CMD_HELP: Starts, stops, or aborts the measurement:

- INITiate... starts or restarts the measurement. The measurement enters
↳ the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY
↳ ' state. Measurement results are kept. The resources remain allocated to the
↳ measurement.
- ABORT... halts the measurement immediately. The measurement enters the
↳ 'OFF' state. All measurement values are set to NAV. Allocated resources are
↳ released.
```

Use FETCh...STATe? to query the current measurement state.

initiate_with_opc() → None

```
# SCPI: INITiate:CDMA:MEASurement<Instance>:OLTR
driver.oltr.initiate_with_opc()

INTRO_CMD_HELP: Starts, stops, or aborts the measurement:

- INITiate... starts or restarts the measurement. The measurement enters
↳ the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY
↳ ' state. Measurement results are kept. The resources remain allocated to the
↳ measurement.
- ABORT... halts the measurement immediately. The measurement enters the
↳ 'OFF' state. All measurement values are set to NAV. Allocated resources are
↳ released.
```

Use FETCh...STATe? to query the current measurement state.

Same as initiate, but waits for the operation to complete before continuing further. Use the RsCmwCdma2kMeas.utilities.opc_timeout_set() to set the timeout value.

stop() → None

```
# SCPI: STOP:CDMA:MEASurement<Instance>:OLTR
driver.oltr.stop()

INTRO_CMD_HELP: Starts, stops, or aborts the measurement:

- INITiate... starts or restarts the measurement. The measurement enters
↳ the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY
↳ ' state. Measurement results are kept. The resources remain allocated to the
↳ measurement.
(continues on next page)
```

(continued from previous page)

```
- ABORt... halts the measurement immediately. The measurement enters the
↳ 'OFF' state. All measurement values are set to NAV. Allocated resources are
↳ released.
```

Use FETCh...STATe? to query the current measurement state.

stop_with_opc() → None

```
# SCPI: STOP:CDMA:MEASurement<Instance>:OLTR
driver.oltr.stop_with_opc()
```

INTRO_CMD_HELP: Starts, stops, or aborts the measurement:

```
- INITiate... starts or restarts the measurement. The measurement enters
↳ the 'RUN' state.
- STOP... halts the measurement immediately. The measurement enters the 'RDY
↳ ' state. Measurement results are kept. The resources remain allocated to the
↳ measurement.
- ABORt... halts the measurement immediately. The measurement enters the
↳ 'OFF' state. All measurement values are set to NAV. Allocated resources are
↳ released.
```

Use FETCh...STATe? to query the current measurement state.

Same as stop, but waits for the operation to complete before continuing further. Use the RsCmwCdma2kMeas.utilities.opc_timeout_set() to set the timeout value.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.oltr.clone()
```

Subgroups

7.5.1 State

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:OLTR:STATe
```

class State

State commands group definition. 2 total commands, 1 Sub-groups, 1 group commands

fetch() → RsCmwCdma2kMeas.enums.ResourceState

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:OLTR:STATe
value: enums.ResourceState = driver.oltr.state.fetch()
```

Queries the main measurement state. Use `FETCh:...:STATe:ALL?` to query the measurement state including the substates. Use `INITiate...`, `STOP...`, `ABORT...` to change the measurement state.

return meas_state: OFF | RUN | RDY OFF: measurement switched off, no resources allocated, no results available (when entered after `ABORT...`) RUN: measurement running (after `INITiate...`, `READ...`) , synchronization pending or adjusted, resources active or queued RDY: measurement has been terminated, valid results are available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.oltr.state.clone()
```

Subgroups

7.5.1.1 All

SCPI Commands

```
FETCh:CDMA:MEASurement<Instance>:OLTR:STATe:ALL
```

class All

All commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class FetchStruct

Response structure. Fields:

- Main_State: enums.ResourceState: OFF | RDY | RUN OFF: measurement switched off, no resources allocated, no results available (when entered after `STOP...`) RDY: measurement has been terminated, valid results are available RUN: measurement running (after `INITiate...`, `READ...`) , synchronization pending or adjusted, resources active or queued
- Sync_State: enums.ResourceState: PEND | ADJ | INV PEND: waiting for resource allocation, adjustment, hardware switching ('pending') ADJ: all necessary adjustments finished, measurement running ('adjusted') INV: not applicable because MainState: OFF or RDY ('invalid')
- Resource_State: enums.ResourceState: QUE | ACT | INV QUE: measurement without resources, no results available ('queued') ACT: resources allocated, acquisition of results in progress but not complete ('active') INV: not applicable because MainState: OFF or RDY ('invalid')

fetch() → FetchStruct

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:OLTR:STATe:ALL
value: FetchStruct = driver.oltr.state.all.fetch()
```

Queries the main measurement state and the measurement substates. Both measurement substates are relevant for running measurements only. Use `FETCh:...:STATe?` to query the main measurement state only. Use `INITiate...`, `STOP...`, `ABORT...` to change the measurement state.

return structure: for return value, see the help for FetchStruct structure arguments.

7.5.2 Sequence<Sequence>

RepCap Settings

```
# Range: Nr1 .. Nr5
rc = driver.oltr.sequence.repcap_sequence_get()
driver.oltr.sequence.repcap_sequence_set(repcap.Sequence.Nr1)
```

class Sequence

Sequence commands group definition. 10 total commands, 1 Sub-groups, 0 group commands Repeated Capability: Sequence, default value after init: Sequence.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.oltr.sequence.clone()
```

Subgroups

7.5.2.1 Trace

class Trace

Trace commands group definition. 10 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.oltr.sequence.trace.clone()
```

Subgroups

7.5.2.1.1 Up

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:UP
FETCh:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:UP
```

class Up

Up commands group definition. 5 total commands, 1 Sub-groups, 2 group commands

fetch(sequence=<Sequence.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:UP
value: List[float] = driver.oltr.sequence.trace.up.fetch(sequence = repcap.
↳ Sequence.Default)
```

Returns the values of the OLTR traces. For each sequence, UP/DOWN commands return the results of the 100 ms interval following the power up/down step. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface 'Sequence')

return up_power: No help available

read(sequence=<Sequence.Default: -1>) → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:UP
value: List[float] = driver.oltr.sequence.trace.up.read(sequence = repcap.
↳ Sequence.Default)
```

Returns the values of the OLTR traces. For each sequence, UP/DOWN commands return the results of the 100 ms interval following the power up/down step. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface 'Sequence')

return up_power: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.oltr.sequence.trace.up.clone()
```

Subgroups

7.5.2.1.1.1 State

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:UP:STATe
FETCh:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:UP:STATe
CALCulate:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:UP:STATe
```

class State

State commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate(sequence=<Sequence.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>
↳ :TRACe:UP:STATe
value: List[enums.ResultStatus2] = driver.oltr.sequence.trace.up.state.
↳ calculate(sequence = repcap.Sequence.Default)
```

(continues on next page)

(continued from previous page)

For each sequence, state commands return limit violation results ‘State Down/Up Power’ in 20 equidistant parts of the 100 ms interval following the power down/up step. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return state_up_power: No help available

fetch(sequence=<Sequence.Default: -1>) → List[RsCmwCdma2kMeas.enums.StatePower]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:UP:STATe
value: List[enums.StatePower] = driver.oltr.sequence.trace.up.state.
↪ fetch(sequence = repcap.Sequence.Default)
```

For each sequence, state commands return limit violation results ‘State Down/Up Power’ in 20 equidistant parts of the 100 ms interval following the power down/up step. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return state_up_power: No help available

read(sequence=<Sequence.Default: -1>) → List[RsCmwCdma2kMeas.enums.StatePower]

```
# SCPI: READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:UP:STATe
value: List[enums.StatePower] = driver.oltr.sequence.trace.up.state.
↪ read(sequence = repcap.Sequence.Default)
```

For each sequence, state commands return limit violation results ‘State Down/Up Power’ in 20 equidistant parts of the 100 ms interval following the power down/up step. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return state_up_power: No help available

7.5.2.1.2 Down

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:DOWN
FETCh:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:DOWN
```

class Down

Down commands group definition. 5 total commands, 1 Sub-groups, 2 group commands

fetch(sequence=<Sequence.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:DOWN
value: List[float] = driver.oltr.sequence.trace.down.fetch(sequence = repcap.
↳Sequence.Default)
```

Returns the values of the OLTR traces. For each sequence, UP/DOWN commands return the results of the 100 ms interval following the power up/down step. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface 'Sequence')

return down_power: No help available

read(sequence=<Sequence.Default: -1>) → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:DOWN
value: List[float] = driver.oltr.sequence.trace.down.read(sequence = repcap.
↳Sequence.Default)
```

Returns the values of the OLTR traces. For each sequence, UP/DOWN commands return the results of the 100 ms interval following the power up/down step. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface 'Sequence')

return down_power: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.oltr.sequence.trace.down.clone()
```

Subgroups

7.5.2.1.2.1 State

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:DOWN:STATe
FETCh:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:DOWN:STATe
CALCulate:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:DOWN:STATe
```

class State

State commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate(sequence=<Sequence.Default: -1>) → List[RsCmwCdma2kMeas.enums.ResultStatus2]

```
# SCPI: CALCulate:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>
↳:TRACe:DOWN:STATe
value: List[enums.ResultStatus2] = driver.oltr.sequence.trace.down.state.
↳calculate(sequence = repcap.Sequence.Default)
```

For each sequence, state commands return limit violation results ‘State Down/Up Power’ in 20 equidistant parts of the 100 ms interval following the power down/up step. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return state_down_power: No help available

fetch(sequence=<Sequence.Default: -1>) → List[RsCmwCdma2kMeas.enums.StatePower]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>
↳:TRACe:DOWN:STATe
value: List[enums.StatePower] = driver.oltr.sequence.trace.down.state.
↳fetch(sequence = repcap.Sequence.Default)
```

For each sequence, state commands return limit violation results ‘State Down/Up Power’ in 20 equidistant parts of the 100 ms interval following the power down/up step. The values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return state_down_power: No help available

read(sequence=<Sequence.Default: -1>) → List[RsCmwCdma2kMeas.enums.StatePower]

```
# SCPI: READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACe:DOWN:STATe
value: List[enums.StatePower] = driver.oltr.sequence.trace.down.state.
↪read(sequence = repcap.Sequence.Default)
```

For each sequence, state commands return limit violation results ‘State Down/Up Power’ in 20 equidistant parts of the 100 ms interval following the power down/up step. The values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each result listed below.

Use RsCmwCdma2kMeas.reliability.last_value to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return state_down_power: No help available

7.6 RpInterval

class RpInterval

RpInterval commands group definition. 4 total commands, 1 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.rpInterval.clone()
```

Subgroups

7.6.1 Sequence<Sequence>

RepCap Settings

```
# Range: Nr1 .. Nr5
rc = driver.rpInterval.sequence.repcap_sequence_get()
driver.rpInterval.sequence.repcap_sequence_set(repcap.Sequence.Nr1)
```

class Sequence

Sequence commands group definition. 4 total commands, 1 Sub-groups, 0 group commands Repeated Capability: Sequence, default value after init: Sequence.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.rpInterval.sequence.clone()
```

Subgroups

7.6.1.1 Trace

class Trace

Trace commands group definition. 4 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.rpInterval.sequence.trace.clone()
```

Subgroups

7.6.1.1.1 Up

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>:TRACe:UP
FETCh:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>:TRACe:UP
```

class Up

Up commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch(sequence=<Sequence.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>:TRACe:UP
value: List[float] = driver.rpInterval.sequence.trace.up.fetch(sequence = ↵
↵repcap.Sequence.Default)
```

Returns the values of the reference power traces. For each sequence, UP/DOWN commands return the results of the reference power interval for the power up/down step. See method `RsCmwCdma2kMeas.Configure.Oltr.RpInterval.time` .

Use `RsCmwCdma2kMeas.reliability.last_value` to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return up_power: No help available

read(sequence=<Sequence.Default: -1>) → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>:TRACe:UP
value: List[float] = driver.rpInterval.sequence.trace.up.read(sequence = repcap.
↵Sequence.Default)
```

(continues on next page)

(continued from previous page)

Returns the values of the reference power traces. For each sequence, UP/DOWN commands return the results of the reference power interval for the power up/down step. See method `RsCmwCdma2kMeas.Configure.Oltr.RpInterval.time` .

Use `RsCmwCdma2kMeas.reliability.last_value` to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return up_power: No help available

7.6.1.1.2 Down

SCPI Commands

```
READ:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>:TRACe:DOWN
FETCh:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>:TRACe:DOWN
```

class Down

Down commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch(sequence=<Sequence.Default: -1>) → List[float]

```
# SCPI: FETCh:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>
↳:TRACe:DOWN
value: List[float] = driver.rpInterval.sequence.trace.down.fetch(sequence =
↳repcap.Sequence.Default)
```

Returns the values of the reference power traces. For each sequence, UP/DOWN commands return the results of the reference power interval for the power up/down step. See method `RsCmwCdma2kMeas.Configure.Oltr.RpInterval.time` .

Use `RsCmwCdma2kMeas.reliability.last_value` to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return down_power: No help available

read(sequence=<Sequence.Default: -1>) → List[float]

```
# SCPI: READ:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>:TRACe:DOWN
value: List[float] = driver.rpInterval.sequence.trace.down.read(sequence =
↳repcap.Sequence.Default)
```

Returns the values of the reference power traces. For each sequence, UP/DOWN commands return the results of the reference power interval for the power up/down step. See method `RsCmwCdma2kMeas.Configure.Oltr.RpInterval.time` .

Use `RsCmwCdma2kMeas.reliability.last_value` to read the updated reliability indicator.

param sequence optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Sequence’)

return down_power: No help available

INDEX

A

ABORT:CDMA:MEASurement<Instance>:MEvaluation,
84
ABORT:CDMA:MEASurement<Instance>:OLTR, 465

C

CALCulate:CDMA:MEASurement<Instance>:MEvaluation:ACP,
184
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:ACP:AVERAGE,
186
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:ACP:CURRENT,
185
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:ACP:MAXimum,
188
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM<AcpMinus>:AVERAGE,
321
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM<AcpMinus>:CURRENT,
320
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM<AcpMinus>:MAXimum,
322
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPP<AcpPlus>:AVERAGE,
339
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPP<AcpPlus>:CURRENT,
338
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPP<AcpPlus>:MAXimum,
340
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:AVERAGE,
352
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:CURRENT,
349
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:ACPM<AcpMinus>:AVERAGE,
324
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:ACPM<AcpMinus>:CURRENT,
323
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:ACPM<AcpMinus>:MAXimum,
325
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:ACPP<AcpPlus>:AVERAGE,
328
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:ACPP<AcpPlus>:CURRENT,
327
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTended:ACPP<AcpPlus>:MAXimum,
329

CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
331
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
330
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
333
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
335
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
336
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
355
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
358
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
342
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
341
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
343
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
344
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
344
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
360
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
346
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
345
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
347
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
348
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:
348
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:A
402
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:C
400
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:M
404

CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:MEASurement<Instance>:MEvaluation:LIST:MODU	
458	201
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:MEASurement<Instance>:MEvaluation:LIST:MODU	
462	201
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
444	196
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:SURrent<Instance>:MEvaluation:LIST:MODU	
443	196
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:MAXiment<Instance>:MEvaluation:LIST:MODU	
445	197
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
441	198
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:SURrent<Instance>:MEvaluation:LIST:MODU	
440	225
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:MAXiment<Instance>:MEvaluation:LIST:MODU	
442	224
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
447	225
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:SURrent<Instance>:MEvaluation:LIST:MODU	
446	226
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:MAXiment<Instance>:MEvaluation:LIST:MODU	
448	221
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
450	220
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:SURrent<Instance>:MEvaluation:LIST:MODU	
449	222
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:MAXiment<Instance>:MEvaluation:LIST:MODU	
451	223
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
438	218
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:SURrent<Instance>:MEvaluation:LIST:MODU	
438	217
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:MAXiment<Instance>:MEvaluation:LIST:MODU	
439	218
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
456	219
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
455	250
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
457	207
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
453	206
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
452	208
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
454	209
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
248	204
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
245	203
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
200	204
CALCulate:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTDMA:CDMA:OVERage<Instance>:MEvaluation:LIST:MODU	
199	205

[illegible]

484 Index


```

CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult::CDMA:MEASurement<Instance>:RFSettings:BClass,
58 41
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult::CDMA:MEASurement<Instance>:RFSettings:CHANnel,
58 41
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult::CDMA:MEASurement<Instance>:RFSettings:EATTenuati
58 41
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult::CDMA:MEASurement<Instance>:RFSettings:ENPower,
58 41
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult::CDMA:MEASurement<Instance>:RFSettings:FOFFset,
58 41
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult::CDMA:MEASurement<Instance>:RFSettings:FREquency,
58 41
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult::CDMA:MEASurement<Instance>:RFSettings:UMARgin,
58 41
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:OBW,
58 41
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CDMA:MEASurement<Instance>:MEvaluation:ACP,
58 184
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CDMA:MEASurement<Instance>:MEvaluation:ACP:AVERage,
58 186
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CDMA:MEASurement<Instance>:MEvaluation:ACP:CURRent,
45 185
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CDMA:MEASurement<Instance>:MEvaluation:ACP:MAXimum,
48 188
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM
48 321
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM
45 320
CONFIGure:CDMA:MEASurement<Instance>:MEvaluation:RESult:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM
45 322
CONFIGure:CDMA:MEASurement<Instance>:OLTR:GINTerval;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM
78 339
CONFIGure:CDMA:MEASurement<Instance>:OLTR:GINTerval;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM
78 338
CONFIGure:CDMA:MEASurement<Instance>:OLTR:LIMitTOWard;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:ACPM
78 340
CONFIGure:CDMA:MEASurement<Instance>:OLTR:MOEXcept;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:AVEr
74 352
CONFIGure:CDMA:MEASurement<Instance>:OLTR:PSTep;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:CURR
76 349
CONFIGure:CDMA:MEASurement<Instance>:OLTR:PSTep;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTe
76 324
CONFIGure:CDMA:MEASurement<Instance>:OLTR:REPetition;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTe
74 323
CONFIGure:CDMA:MEASurement<Instance>:OLTR:RPINterval;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTe
77 325
CONFIGure:CDMA:MEASurement<Instance>:OLTR:RPINterval;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTe
77 328
CONFIGure:CDMA:MEASurement<Instance>:OLTR:SEQUence;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTe
74 327
CONFIGure:CDMA:MEASurement<Instance>:OLTR:TOUT;CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTe
74 329
CONFIGure:CDMA:MEASurement<Instance>:RCONfig, FETCh:CDMA:MEASurement<Instance>:MEvaluation:LIST:ACP:EXTe
39 331

```

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RCCCH:CDMA:EXTMEASur:Current<Instance>:MEvaluation:LIST:CP:RCCCH:330 381
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RCCCH:CDMA:EXTMEASur:Maximum<Instance>:MEvaluation:LIST:CP:RCCCH:333 379
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RCCCH:CDMA:EXTMEASur:Minimum<Instance>:MEvaluation:LIST:CP:RCCCH:335 378
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RCCCH:CDMA:EXTMEASur:SDV:all<Instance>:MEvaluation:LIST:CP:RCCCH:336 380
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RCCCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RCCCH:355 380
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RCCCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RCCCH:358 378
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:CDMA:POWERMEASur:Ravg<Instance>:MEvaluation:LIST:CP:REACH:342 386
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:CDMA:POWERMEASur:Rmin<Instance>:MEvaluation:LIST:CP:REACH:341 385
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:CDMA:POWERMEASur:Rmax<Instance>:MEvaluation:LIST:CP:REACH:343 387
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:CDMA:POWERMEASur:Rmean<Instance>:MEvaluation:LIST:CP:REACH:344 387
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:REACH:CDMA:POWERMEASur:SE:mean<Instance>:MEvaluation:LIST:CP:REACH:344 385
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RFCH:CDMA:OTOMEASur:Success<Instance>:MEvaluation:LIST:CP:RFCH:319 390
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RFCH:CDMA:SEMEASur:Success<Instance>:MEvaluation:LIST:CP:RFCH:360 389
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RFCH:CDMA:STOMEASur:Success<Instance>:MEvaluation:LIST:CP:RFCH:319 390
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RFCH:CDMA:POWERMEASur:Ravg<Instance>:MEvaluation:LIST:CP:RFCH:346 391
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RFCH:CDMA:POWERMEASur:Rmin<Instance>:MEvaluation:LIST:CP:RFCH:345 388
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RPICH:CDMA:POWERMEASur:Rmin<Instance>:MEvaluation:LIST:CP:RPICH:347 375
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RPICH:CDMA:POWERMEASur:Rmax<Instance>:MEvaluation:LIST:CP:RPICH:348 374
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RPICH:CDMA:POWERMEASur:SE:mean<Instance>:MEvaluation:LIST:CP:RPICH:348 376
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RPICH:CDMA:POWERMEASur:Ravg<Instance>:MEvaluation:LIST:CP:RPICH:402 377
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RPICH:CDMA:POWERMEASur:Rmin<Instance>:MEvaluation:LIST:CP:RPICH:400 374
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:CDMA:IMEASur:Success<Instance>:MEvaluation:LIST:CP:RSCH:404 397
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:CDMA:IMEASur:Success<Instance>:MEvaluation:LIST:CP:RSCH:406 397
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:CDMA:POWERMEASur:Ravg<Instance>:MEvaluation:LIST:CP:RSCH:382 398
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:CDMA:POWERMEASur:Rmin<Instance>:MEvaluation:LIST:CP:RSCH:382 399
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:CDMA:POWERMEASur:Rmax<Instance>:MEvaluation:LIST:CP:RSCH:383 396
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP:RSCH:CDMA:POWERMEASur:Rmean<Instance>:MEvaluation:LIST:CP:RSCH:384 394

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RPIQ
 393 409
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RSCH
 394 428
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RSCH
 395 427
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RSCH
 392 429
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RSCH
 408 427
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RSCH
 432 425
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RSCH
 430 424
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RSCH
 434 426
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:RSCH
 416 424
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CP0:STAT
 415 436
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:AVEF
 417 460
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:CURF
 415 458
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:MAXI
 413 462
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RCCO
 412 444
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RCCO
 414 443
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RCCO
 412 445
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RCCO
 419 443
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RDCC
 418 441
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RDCC
 420 440
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RDCC
 418 442
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RDCC
 422 440
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REAC
 421 447
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REAC
 423 446
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REAC
 421 448
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:REAC
 410 446
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RFCH
 410 450
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:CDMA:MEASurement<Instance>:MEvaluation:LIST:CTO:RFCH
 411 449

Index	489
--------------	------------

FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:CD:CURRENT;MEvaluation:TRACE:ACP:EXT
 311 102
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:CD:MAXimum;MEvaluation:TRACE:ACP:EXT
 315 101
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:CD:STATE;MEvaluation:TRACE:ACP:EXT
 317 99
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:MOD:duration;MEvaluation:TRACE:ACP:EXT
 261 98
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:MOD:duration;CURReputation:TRACE:ACP:EXT
 258 105
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:MOD:duration;MAXimum:TRACE:ACP:EXT
 263 104
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:MOD:duration;MINimum:TRACE:ACP:MAX
 266 111
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:MOD:duration;SDEvaluation:TRACE:ACP:MAX
 268 110
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:OBLin:Average;MEvaluation:TRACE:CDE:ISI
 290 128
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:OBLin:CURRENT;MEvaluation:TRACE:CDE:ISI
 289 127
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:OBLin:MAXimum;MEvaluation:TRACE:CDE:ISI
 292 131
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:OBLin:SDV:direction;MEvaluation:TRACE:CDE:ISI
 293 129
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:LIST:SECDMA:MEASurement<Instance>:MEvaluation:TRACE:CDE:ISI
 193 130
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:MOD:CDMA:Average;MEASurement<Instance>:MEvaluation:TRACE:CDE:QSI
 174 132
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:MOD:CDMA:CURReputation;MEASurement<Instance>:MEvaluation:TRACE:CDE:QSI
 171 131
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:MOD:CDMA:MAXimum;MEASurement<Instance>:MEvaluation:TRACE:CDE:QSI
 177 135
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:MOD:CDMA:MINimum;MEASurement<Instance>:MEvaluation:TRACE:CDE:QSI
 179 134
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:MOD:CDMA:SDV:direction;MEASurement<Instance>:MEvaluation:TRACE:CDE:QSI
 181 135
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:OBW:Average;MEASurement<Instance>:MEvaluation:TRACE:CDP:ISI
 190 116
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:OBW:CURRENT;MEASurement<Instance>:MEvaluation:TRACE:CDP:ISI
 189 115
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:OBW:MAXimum;MEASurement<Instance>:MEvaluation:TRACE:CDP:ISI
 192 120
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:SHATCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDP:ISI
 86 117
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:SHATCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDP:ISI
 87 119
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRATCH:CDMA:MEASurement<Instance>:ABSolute;MEvaluation:TRACE:CDP:ISI
 108 120
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRATCH:CDMA:MEASurement<Instance>:Relative;MEvaluation:TRACE:CDP:QSI
 107 122
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRATCH:CDMA:MEASurement<Instance>:ABSolute;MEvaluation:TRACE:CDP:QSI
 96 121
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRATCH:CDMA:MEASurement<Instance>:Relative;MEvaluation:TRACE:CDP:QSI
 95 126

FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:MAXimum, 123 168
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:MINimum, 124 169
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:STATE, 125 89
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:AVERage, 137 88
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:CURRENT, 136 89
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:MAXimum, 139 170
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:MINimum, 140 91
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:STATE, 141 90
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:AVERage, 143 92
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:CURRENT, 142 112
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:MAXimum, 145 93
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:MINimum, 147 93
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:STATE, 148 94
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:AVERage, 151 114
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:CURRENT, 149 472
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:MAXimum, 152 473
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:STATE, 153 469
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:AVERage, 156 470
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:CURRENT, 154 467
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:MAXimum, 157 468
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:STATE, 159 476
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:AVERage, 161 475
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:ISignal:CURRENT, 160
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:ISignal:MAXimum, 163 84
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:ISignal:STATE, 164 465
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:AVERage, 166
 READ:CDMA:MEASurement<Instance>:MEvaluation:ACP, 184
 FETCH:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSignal:CURRENT, 165

READ:CDMA:MEASurement<Instance>:MEvaluation:ACPEAVERAGE;MEASurement<Instance>:MEvaluation:TRACE:CDE:QSIG
 186 131
 READ:CDMA:MEASurement<Instance>:MEvaluation:ACPECDRCDMA;MEASurement<Instance>:MEvaluation:TRACE:CDE:QSIG
 185 134
 READ:CDMA:MEASurement<Instance>:MEvaluation:ACPEMAXCDMA;MEASurement<Instance>:MEvaluation:TRACE:CDP:ISIG
 188 116
 READ:CDMA:MEASurement<Instance>:MEvaluation:MOBILITYCDMA:MEASurement<Instance>:MEvaluation:TRACE:CDP:ISIG
 174 115
 READ:CDMA:MEASurement<Instance>:MEvaluation:MOBILITYCDMA:MEASurement<Instance>:MEvaluation:TRACE:CDP:ISIG
 171 117
 READ:CDMA:MEASurement<Instance>:MEvaluation:MOBILITYCDMA:MEASurement<Instance>:MEvaluation:TRACE:CDP:ISIG
 177 119
 READ:CDMA:MEASurement<Instance>:MEvaluation:MOBILITYCDMA:MEASurement<Instance>:MEvaluation:TRACE:CDP:QSIG
 179 122
 READ:CDMA:MEASurement<Instance>:MEvaluation:MOBILITYCDMA:MEASurement<Instance>:MEvaluation:TRACE:CDP:QSIG
 181 121
 READ:CDMA:MEASurement<Instance>:MEvaluation:OBPEAVERAGE;MEASurement<Instance>:MEvaluation:TRACE:CDP:QSIG
 190 123
 READ:CDMA:MEASurement<Instance>:MEvaluation:OBPECDRCDMA;MEASurement<Instance>:MEvaluation:TRACE:CDP:QSIG
 189 124
 READ:CDMA:MEASurement<Instance>:MEvaluation:OBPEMAXCDMA;MEASurement<Instance>:MEvaluation:TRACE:CP:ISIG
 192 137
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CP:ISIG
 108 136
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CP:ISIG
 107 139
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CP:ISIG
 96 140
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CP:QSIG
 95 143
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:EXTENDEDMEASUREMENTS:ABSOLUTE
 102 142
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:EXTENDEDMEASUREMENTS:RELATIVE
 101 145
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:EXTENDEDMEASUREMENTS:ABSOLUTE
 99 147
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:EXTENDEDMEASUREMENTS:RELATIVE
 98 151
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:EXTENDEDMEASUREMENTS:ABSOLUTE
 105 149
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:EXTENDEDMEASUREMENTS:RELATIVE
 104 152
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASUREMENTS:ABSOLUTE
 111 156
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASUREMENTS:RELATIVE
 110 154
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASUREMENTS:ABSOLUTE
 128 157
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASUREMENTS:RELATIVE
 127 161
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASUREMENTS:ABSOLUTE
 129 160
 READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CDMA:MEASUREMENTS:RELATIVE
 132 163

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSSignal:AVERage, 465
166

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSSignal:CURRENT, 165
165

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:CTO:QSSignal:MAXimum, 168
168

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:EVMagnitude:AVERage, 89
89

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:EVMagnitude:CURRENT, 88
88

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:EVMagnitude:MAXimum, 89
89

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:IQ:CURRENT, 170
170

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:MERRor:AVERage, 91
91

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:MERRor:CURRENT, 90
90

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:MERRor:MAXimum, 92
92

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:OBW:CURRENT, 112
112

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:PERRor:AVERage, 93
93

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:PERRor:CURRENT, 93
93

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:PERRor:MAXimum, 94
94

READ:CDMA:MEASurement<Instance>:MEvaluation:TRACE:SPECTrum:CURRENT, 114
114

READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACE:DOWN, 472
472

READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACE:DOWN:STATE, 473
473

READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACE:UP, 469
469

READ:CDMA:MEASurement<Instance>:OLTR:SEquence<Sequence>:TRACE:UP:STATE, 470
470

READ:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>:TRACE:DOWN, 476
476

READ:CDMA:MEASurement<Instance>:RPInterval:SEquence<Sequence>:TRACE:UP, 475
475

ROUTE:CDMA:MEASurement<Instance>, 37

ROUTE:CDMA:MEASurement<Instance>:SCENario, 37

ROUTE:CDMA:MEASurement<Instance>:SCENario:CATalog:CSPath, 39
39

ROUTE:CDMA:MEASurement<Instance>:SCENario:CSPath, 37
37

ROUTE:CDMA:MEASurement<Instance>:SCENario:SALone, 37
37

S

STOP:CDMA:MEASurement<Instance>:MEvaluation, 84
84