
RsCmwWcdmaMeas

Release 3.8.10.5

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	AclrMode	23
3.2	ActiveLimit	23
3.3	AnalysisMode	23
3.4	AutoManualMode	23
3.5	Band	24
3.6	Carrier	24
3.7	CmwsConnector	24
3.8	DetectionMode	25
3.9	LimitHmode	25
3.10	MeasMode	25
3.11	MeasPeriod	25
3.12	Mode	25
3.13	Modulation	26
3.14	OutPowFstate	26
3.15	ParameterSetMode	26
3.16	PatternType	26
3.17	PcdErrorPhase	26
3.18	Repeat	27
3.19	ResourceState	27
3.20	ResultStatus2	27
3.21	Retrigger	27
3.22	RxConnector	27
3.23	RxConverter	28
3.24	SetType	28

3.25	SignalSlope	29
3.26	SlotNumber	29
3.27	SpreadingFactorA	29
3.28	SpreadingFactorB	29
3.29	State	29
3.30	StopCondition	30
3.31	TestCase	30
3.32	TestScenarioB	30
3.33	Type	30
3.34	UlConfiguration	30
4	RepCaps	31
4.1	Instance (Global)	31
4.2	Carrier	31
4.3	CARRierExt	31
4.4	EdpdChannel	32
4.5	Minus	32
4.6	Plus	32
4.7	Preamble	32
4.8	RampUpCarrier	32
4.9	Segment	33
5	Examples	35
6	Index	37
7	RsCmwWcdmaMeas API Structure	39
7.1	Route	41
7.1.1	Scenario	41
7.1.1.1	MaProtocol	43
7.2	Configure	43
7.2.1	Carrier<Carrier>	43
7.2.1.1	Band	44
7.2.2	Cell	45
7.2.2.1	Carrier<Carrier>	45
7.2.2.1.1	Scode	45
7.2.3	UeSignal	46
7.2.3.1	Carrier<Carrier>	49
7.2.3.1.1	Scode	49
7.2.4	UeChannels	50
7.2.4.1	Carrier<Carrier>	51
7.2.4.1.1	Edpdch<EdpdChannel>	51
7.2.4.1.2	Edpcch	53
7.2.4.1.3	Hsdpcch	54
7.2.4.1.3.1	Config	55
7.2.4.1.4	Dpdch	56
7.2.4.1.5	Dpcch	57
7.2.5	RfSettings	59
7.2.5.1	Dcarrier	61
7.2.5.2	Carrier<Carrier>	61
7.2.5.2.1	Frequency	62
7.2.6	MultiEval	63
7.2.6.1	Scount	66
7.2.6.2	Limit	68
7.2.6.2.1	RcdError	72

	7.2.6.2.1.1	Eecdp	73
	7.2.6.2.1.2	Carrier<Carrier>	73
	7.2.6.2.1.3	Edpdch<EdpdChannel>	77
	7.2.6.2.1.4	Edpcch	79
	7.2.6.2.1.5	Hsdpcch	80
	7.2.6.2.1.6	Config	81
	7.2.6.2.1.7	Dpdch	82
	7.2.6.2.1.8	Dpcch	83
	7.2.6.2.2	Pcontrol	84
	7.2.6.2.3	Emask	85
	7.2.6.2.3.1	Dcarrier	87
	7.2.6.2.4	Aclr	88
7.2.6.3		Rotation	89
7.2.6.4		DsFactor	90
7.2.6.5		Sscalar	90
7.2.6.6		CdThreshold	91
7.2.6.7		Dmode	92
7.2.6.8		Amode	92
7.2.6.9		Mperiod	93
7.2.6.10		Result	93
	7.2.6.10.1	Chip	101
7.2.6.11		ListPy	103
	7.2.6.11.1	Segment<Segment>	105
	7.2.6.11.1.1	Setup	105
	7.2.6.11.1.2	Modulation	106
	7.2.6.11.1.3	Spectrum	107
	7.2.6.11.1.4	CdPower	108
	7.2.6.11.1.5	UePower	109
	7.2.6.11.1.6	Phd	110
	7.2.6.11.1.7	SingleCmw	111
	7.2.6.11.1.8	Connector	111
	7.2.6.11.2	SingleCmw	112
7.2.7		Tpc	113
	7.2.7.1	IlpControl	115
	7.2.7.2	Monitor	118
	7.2.7.3	Mpedch	118
	7.2.7.4	Ctfc	119
	7.2.7.4.1	Mlength	119
	7.2.7.5	Ulcmm	120
	7.2.7.6	Dhib	121
	7.2.7.7	Limit	122
	7.2.7.7.1	IlpControl	124
	7.2.7.7.1.1	MaxPower	127
	7.2.7.7.2	Ulcmm	129
7.2.8		Prach	131
	7.2.8.1	Limit	133
	7.2.8.1.1	Pcontrol	136
	7.2.8.1.1.1	MaxPower	138
	7.2.8.2	Result	141
	7.2.8.2.1	Chip	145
7.2.9		OoSync	147
	7.2.9.1	Limit	148
7.2.10		OlpControl	149
7.3		Trigger	151

7.3.1	MultiEval	151
7.3.1.1	Catalog	154
7.3.1.2	ListPy	154
7.3.2	Tpc	155
7.3.2.1	Catalog	158
7.3.3	Prach	159
7.3.3.1	Catalog	161
7.3.4	OoSync	162
7.3.4.1	Catalog	165
7.3.5	OlpControl	165
7.3.5.1	Catalog	168
7.4	MultiEval	168
7.4.1	State	171
7.4.1.1	All	172
7.4.2	Trace	172
7.4.2.1	Phd	173
7.4.2.1.1	Current	173
7.4.2.2	CdeMonitor	174
7.4.2.2.1	Qsignal	174
7.4.2.2.1.1	Current	175
7.4.2.2.2	Isignal	176
7.4.2.2.2.1	Current	176
7.4.2.3	CdpMonitor	177
7.4.2.3.1	Qsignal	177
7.4.2.3.1.1	Current	177
7.4.2.3.2	Isignal	178
7.4.2.3.2.1	Current	178
7.4.2.4	Carrier<Carrier>	179
7.4.2.4.1	Perror	179
7.4.2.4.1.1	Rms	180
7.4.2.4.1.2	Current	180
7.4.2.5	Emask	181
7.4.2.5.1	MfLeft	181
7.4.2.5.1.1	Average	181
7.4.2.5.1.2	Current	182
7.4.2.5.1.3	Maximum	183
7.4.2.5.2	MfRight	184
7.4.2.5.2.1	Average	184
7.4.2.5.2.2	Current	186
7.4.2.5.2.3	Maximum	187
7.4.2.5.3	HkfLeft	188
7.4.2.5.3.1	Average	188
7.4.2.5.3.2	Current	189
7.4.2.5.3.3	Maximum	190
7.4.2.5.4	HkfRight	191
7.4.2.5.4.1	Average	191
7.4.2.5.4.2	Current	192
7.4.2.5.4.3	Maximum	193
7.4.2.5.5	Kfilter	194
7.4.2.5.5.1	Current	194
7.4.2.5.5.2	Average	195
7.4.2.5.5.3	Maximum	196
7.4.2.6	EvMagnitude	197
7.4.2.6.1	Chip	197

	7.4.2.6.1.1	Maximum	197
	7.4.2.6.1.2	Average	198
	7.4.2.6.1.3	Current	199
7.4.2.7	Merror		199
	7.4.2.7.1	Chip	200
	7.4.2.7.1.1	Maximum	200
	7.4.2.7.1.2	Average	201
	7.4.2.7.1.3	Current	202
7.4.2.8	Perror		202
	7.4.2.8.1	Chip	203
	7.4.2.8.1.1	Maximum	203
	7.4.2.8.1.2	Average	204
	7.4.2.8.1.3	Current	204
7.4.2.9	Iq		205
	7.4.2.9.1	Current	205
7.4.3	Carrier<Carrier>		206
	7.4.3.1	Trace	206
	7.4.3.1.1	UePower	207
	7.4.3.1.1.1	StandardDev	207
	7.4.3.1.1.2	Minimum	208
	7.4.3.1.1.3	Maximum	209
	7.4.3.1.1.4	Average	210
	7.4.3.1.1.5	Current	211
	7.4.3.1.2	EvMagnitude	212
	7.4.3.1.2.1	Rms	212
	7.4.3.1.2.2	Sdeviaton	213
	7.4.3.1.2.3	StandardDev	213
	7.4.3.1.2.4	Maximum	214
	7.4.3.1.2.5	Average	215
	7.4.3.1.2.6	Current	216
	7.4.3.1.2.7	Peak	217
	7.4.3.1.2.8	Maximum	217
	7.4.3.1.2.9	StandardDev	218
	7.4.3.1.2.10	Average	219
	7.4.3.1.2.11	Sdeviaton	220
	7.4.3.1.2.12	Current	220
	7.4.3.1.3	Merror	221
	7.4.3.1.3.1	Rms	221
	7.4.3.1.3.2	StandardDev	222
	7.4.3.1.3.3	Maximum	223
	7.4.3.1.3.4	Average	224
	7.4.3.1.3.5	Current	225
	7.4.3.1.3.6	Peak	226
	7.4.3.1.3.7	StandardDev	226
	7.4.3.1.3.8	Maximum	227
	7.4.3.1.3.9	Average	228
	7.4.3.1.3.10	Current	229
	7.4.3.1.4	Perror	230
	7.4.3.1.4.1	Rms	230
	7.4.3.1.4.2	StandardDev	230
	7.4.3.1.4.3	Maximum	231
	7.4.3.1.4.4	Average	232
	7.4.3.1.4.5	Current	233
	7.4.3.1.4.6	Peak	234

7.4.3.1.4.7	StandardDev	234
7.4.3.1.4.8	Maximum	235
7.4.3.1.4.9	Average	236
7.4.3.1.4.10	Current	237
7.4.3.1.5	CdPower	238
7.4.3.1.5.1	Dpcch	238
7.4.3.1.5.2	Current	238
7.4.3.1.5.3	Average	239
7.4.3.1.5.4	Minimum	240
7.4.3.1.5.5	Maximum	241
7.4.3.1.5.6	StandardDev	242
7.4.3.1.5.7	Dpdch	243
7.4.3.1.5.8	Current	243
7.4.3.1.5.9	Average	244
7.4.3.1.5.10	Minimum	245
7.4.3.1.5.11	Maximum	246
7.4.3.1.5.12	StandardDev	247
7.4.3.1.5.13	Hsdpcch	249
7.4.3.1.5.14	Current	249
7.4.3.1.5.15	Average	250
7.4.3.1.5.16	Minimum	251
7.4.3.1.5.17	Maximum	252
7.4.3.1.5.18	StandardDev	253
7.4.3.1.5.19	Edpcch	254
7.4.3.1.5.20	Current	254
7.4.3.1.5.21	Average	255
7.4.3.1.5.22	Minimum	256
7.4.3.1.5.23	Maximum	257
7.4.3.1.5.24	StandardDev	258
7.4.3.1.5.25	Edpdch<EdpdChannel>	259
7.4.3.1.5.26	Current	260
7.4.3.1.5.27	Average	261
7.4.3.1.5.28	Minimum	262
7.4.3.1.5.29	Maximum	263
7.4.3.1.5.30	StandardDev	264
7.4.3.1.6	CdError	265
7.4.3.1.6.1	Dpcch	266
7.4.3.1.6.2	Current	266
7.4.3.1.6.3	Average	267
7.4.3.1.6.4	Maximum	268
7.4.3.1.6.5	StandardDev	269
7.4.3.1.6.6	Dpdch	270
7.4.3.1.6.7	Current	270
7.4.3.1.6.8	Average	271
7.4.3.1.6.9	Maximum	272
7.4.3.1.6.10	StandardDev	273
7.4.3.1.6.11	Hsdpcch	274
7.4.3.1.6.12	Current	274
7.4.3.1.6.13	Average	275
7.4.3.1.6.14	Maximum	276
7.4.3.1.6.15	StandardDev	277
7.4.3.1.6.16	Edpcch	278
7.4.3.1.6.17	Current	279
7.4.3.1.6.18	Average	280

7.4.3.1.6.19	Maximum	281
7.4.3.1.6.20	StandardDev	282
7.4.3.1.6.21	Edpdch<EdpdChannel>	283
7.4.3.1.6.22	Current	283
7.4.3.1.6.23	Average	284
7.4.3.1.6.24	Maximum	285
7.4.3.1.6.25	StandardDev	287
7.4.3.1.7	FreqError	288
7.4.3.1.7.1	StandardDev	288
7.4.3.1.7.2	Maximum	289
7.4.3.1.7.3	Average	290
7.4.3.1.7.4	Current	291
7.4.3.1.8	Psteps	292
7.4.3.1.8.1	StandardDev	292
7.4.3.1.8.2	Minimum	293
7.4.3.1.8.3	Maximum	294
7.4.3.1.8.4	Average	295
7.4.3.1.8.5	Current	296
7.4.3.1.9	RcdError	297
7.4.3.1.9.1	Sf	298
7.4.3.1.9.2	Dpcch	298
7.4.3.1.9.3	Dpdch	299
7.4.3.1.9.4	Hsdpcch	300
7.4.3.1.9.5	Edpcch	301
7.4.3.1.9.6	Edpdch<EdpdChannel>	302
7.4.3.1.9.7	Dpcch	303
7.4.3.1.9.8	Current	303
7.4.3.1.9.9	Average	304
7.4.3.1.9.10	Maximum	305
7.4.3.1.9.11	StandardDev	306
7.4.3.1.9.12	Dpdch	307
7.4.3.1.9.13	Current	308
7.4.3.1.9.14	Average	309
7.4.3.1.9.15	Maximum	310
7.4.3.1.9.16	StandardDev	311
7.4.3.1.9.17	Hsdpcch	312
7.4.3.1.9.18	Current	312
7.4.3.1.9.19	Average	313
7.4.3.1.9.20	Maximum	314
7.4.3.1.9.21	StandardDev	315
7.4.3.1.9.22	Edpcch	316
7.4.3.1.9.23	Current	316
7.4.3.1.9.24	Average	317
7.4.3.1.9.25	Maximum	318
7.4.3.1.9.26	StandardDev	319
7.4.3.1.9.27	Edpdch<EdpdChannel>	320
7.4.3.1.9.28	Current	321
7.4.3.1.9.29	Average	322
7.4.3.1.9.30	Maximum	323
7.4.3.1.9.31	StandardDev	324
7.4.3.2	Modulation	325
7.4.3.2.1	StandardDev	325
7.4.3.2.2	Maximum	328
7.4.3.2.3	Current	330

7.4.3.2.4	Average	332
7.4.3.3	RcdError	334
7.4.3.3.1	Current	334
7.4.3.3.2	Average	336
7.4.3.3.3	Maximum	338
7.4.3.3.4	StandardDev	340
7.4.3.3.5	Sf	342
7.4.3.3.6	OcInfo	343
7.4.3.4	CdPower	345
7.4.3.4.1	Current	345
7.4.3.4.2	Average	346
7.4.3.4.3	Minimum	348
7.4.3.4.4	Maximum	349
7.4.3.4.5	StandardDev	350
7.4.3.5	CdError	351
7.4.3.5.1	Current	352
7.4.3.5.2	Average	353
7.4.3.5.3	Maximum	354
7.4.3.5.4	StandardDev	355
7.4.4	Spectrum	357
7.4.4.1	Average	357
7.4.4.2	Current	360
7.4.4.3	Maximum	363
7.4.5	Modulation	366
7.4.5.1	Uephd	366
7.4.5.2	PhDhsDpcch	368
7.4.6	Ber	369
7.4.7	Pcde	370
7.4.7.1	Current	370
7.4.7.2	Maximum	371
7.4.8	ListPy	372
7.4.8.1	Sreliability	372
7.4.8.2	UePower	373
7.4.8.2.1	Current	373
7.4.8.3	Segment<Segment>	373
7.4.8.3.1	UePower	374
7.4.8.3.1.1	Current	374
7.4.8.3.2	Phd	375
7.4.8.3.2.1	Current	375
7.4.8.3.3	Pcde	376
7.4.8.3.3.1	Current	376
7.4.8.3.3.2	Maximum	377
7.4.8.3.4	CdPower	377
7.4.8.3.4.1	Current	378
7.4.8.3.4.2	Average	379
7.4.8.3.4.3	Minimum	380
7.4.8.3.4.4	Maximum	381
7.4.8.3.4.5	StandardDev	382
7.4.8.3.5	Spectrum	383
7.4.8.3.5.1	Current	383
7.4.8.3.5.2	Average	384
7.4.8.3.5.3	Maximum	386
7.4.8.3.6	Modulation	387
7.4.8.3.6.1	StandardDev	387

7.4.8.3.6.2	Maximum	388
7.4.8.3.6.3	Current	389
7.4.8.3.6.4	Average	390
7.4.8.3.7	CdError	391
7.4.8.3.7.1	Current	391
7.4.8.3.7.2	Average	392
7.4.8.3.7.3	Maximum	393
7.4.8.3.7.4	StandardDev	394
7.4.8.4	Phd	395
7.4.8.4.1	Current	395
7.4.8.5	Pcde	396
7.4.8.5.1	Code	396
7.4.8.5.1.1	Maximum	396
7.4.8.5.1.2	Current	397
7.4.8.5.2	Phase	397
7.4.8.5.2.1	Maximum	397
7.4.8.5.2.2	Current	398
7.4.8.5.3	Error	398
7.4.8.5.3.1	Maximum	399
7.4.8.5.3.2	Current	399
7.4.8.5.4	Current	400
7.4.8.5.5	Maximum	400
7.4.8.6	CdPower	401
7.4.8.6.1	Dpcch	401
7.4.8.6.1.1	Current	401
7.4.8.6.1.2	Average	402
7.4.8.6.1.3	Minimum	402
7.4.8.6.1.4	Maximum	403
7.4.8.6.1.5	StandardDev	403
7.4.8.6.2	Dpdch	403
7.4.8.6.2.1	Current	404
7.4.8.6.2.2	Average	404
7.4.8.6.2.3	Minimum	405
7.4.8.6.2.4	Maximum	405
7.4.8.6.2.5	StandardDev	405
7.4.8.6.3	Hsdpcch	406
7.4.8.6.3.1	Current	406
7.4.8.6.3.2	Average	407
7.4.8.6.3.3	Minimum	407
7.4.8.6.3.4	Maximum	408
7.4.8.6.3.5	StandardDev	408
7.4.8.6.4	Edpcch	409
7.4.8.6.4.1	Current	409
7.4.8.6.4.2	Average	409
7.4.8.6.4.3	Minimum	410
7.4.8.6.4.4	Maximum	410
7.4.8.6.4.5	StandardDev	411
7.4.8.6.5	Edpdch<EdpdChannel>	411
7.4.8.6.5.1	Current	411
7.4.8.6.5.2	Average	412
7.4.8.6.5.3	Minimum	412
7.4.8.6.5.4	Maximum	413
7.4.8.6.5.5	StandardDev	413
7.4.8.6.6	StandardDev	414

7.4.8.6.7	Maximum	415
7.4.8.6.8	Minimum	416
7.4.8.6.9	Average	417
7.4.8.6.10	Current	418
7.4.8.7	Spectrum	419
7.4.8.7.1	UePower	419
7.4.8.7.1.1	Current	419
7.4.8.7.1.2	Average	420
7.4.8.7.1.3	Maximum	420
7.4.8.7.2	Emask	421
7.4.8.7.2.1	Hda	421
7.4.8.7.2.2	Current	421
7.4.8.7.2.3	Average	422
7.4.8.7.2.4	Maximum	422
7.4.8.7.2.5	Had	423
7.4.8.7.2.6	Current	423
7.4.8.7.2.7	Average	423
7.4.8.7.2.8	Maximum	424
7.4.8.7.2.9	Ab	424
7.4.8.7.2.10	Current	424
7.4.8.7.2.11	Average	425
7.4.8.7.2.12	Maximum	426
7.4.8.7.2.13	Bc	426
7.4.8.7.2.14	Current	426
7.4.8.7.2.15	Average	427
7.4.8.7.2.16	Maximum	427
7.4.8.7.2.17	Cd	428
7.4.8.7.2.18	Current	428
7.4.8.7.2.19	Average	428
7.4.8.7.2.20	Maximum	429
7.4.8.7.2.21	Ef	429
7.4.8.7.2.22	Current	430
7.4.8.7.2.23	Average	430
7.4.8.7.2.24	Maximum	431
7.4.8.7.2.25	Fe	431
7.4.8.7.2.26	Current	431
7.4.8.7.2.27	Average	432
7.4.8.7.2.28	Maximum	432
7.4.8.7.2.29	Dc	433
7.4.8.7.2.30	Current	433
7.4.8.7.2.31	Average	433
7.4.8.7.2.32	Maximum	434
7.4.8.7.2.33	Cb	434
7.4.8.7.2.34	Current	435
7.4.8.7.2.35	Average	435
7.4.8.7.2.36	Maximum	436
7.4.8.7.2.37	Ba	436
7.4.8.7.2.38	Current	436
7.4.8.7.2.39	Average	437
7.4.8.7.2.40	Maximum	437
7.4.8.7.3	Cpower	438
7.4.8.7.3.1	Current	438
7.4.8.7.3.2	Average	438
7.4.8.7.3.3	Maximum	439

7.4.8.7.4	Aclr	439
7.4.8.7.4.1	M<Minus>	440
7.4.8.7.4.2	Current	440
7.4.8.7.4.3	Average	441
7.4.8.7.4.4	Maximum	441
7.4.8.7.4.5	P<Plus>	442
7.4.8.7.4.6	Current	442
7.4.8.7.4.7	Average	443
7.4.8.7.4.8	Maximum	444
7.4.8.7.5	Obw	445
7.4.8.7.5.1	Current	445
7.4.8.7.5.2	Average	445
7.4.8.7.5.3	Maximum	446
7.4.8.7.6	Current	446
7.4.8.7.7	Average	447
7.4.8.7.8	Maximum	449
7.4.8.8	Modulation	450
7.4.8.8.1	Evm	450
7.4.8.8.1.1	Rms	450
7.4.8.8.1.2	Current	451
7.4.8.8.1.3	Average	451
7.4.8.8.1.4	Maximum	452
7.4.8.8.1.5	StandardDev	452
7.4.8.8.1.6	Peak	453
7.4.8.8.1.7	Current	453
7.4.8.8.1.8	Average	453
7.4.8.8.1.9	Maximum	454
7.4.8.8.1.10	StandardDev	454
7.4.8.8.2	Merror	455
7.4.8.8.2.1	Rms	455
7.4.8.8.2.2	Current	455
7.4.8.8.2.3	Average	456
7.4.8.8.2.4	Maximum	456
7.4.8.8.2.5	StandardDev	457
7.4.8.8.2.6	Peak	457
7.4.8.8.2.7	Current	457
7.4.8.8.2.8	Average	458
7.4.8.8.2.9	Maximum	458
7.4.8.8.2.10	StandardDev	459
7.4.8.8.3	Perror	459
7.4.8.8.3.1	Rms	459
7.4.8.8.3.2	Current	460
7.4.8.8.3.3	Average	460
7.4.8.8.3.4	Maximum	461
7.4.8.8.3.5	StandardDev	461
7.4.8.8.3.6	Peak	462
7.4.8.8.3.7	Current	462
7.4.8.8.3.8	Average	462
7.4.8.8.3.9	Maximum	463
7.4.8.8.3.10	StandardDev	463
7.4.8.8.4	IqOffset	464
7.4.8.8.4.1	Current	464
7.4.8.8.4.2	Average	465
7.4.8.8.4.3	Maximum	465

7.4.8.8.4.4	StandardDev	466
7.4.8.8.5	IqImbalance	466
7.4.8.8.5.1	Current	466
7.4.8.8.5.2	Average	467
7.4.8.8.5.3	Maximum	467
7.4.8.8.5.4	StandardDev	468
7.4.8.8.6	FreqError	468
7.4.8.8.6.1	Current	468
7.4.8.8.6.2	Average	469
7.4.8.8.6.3	Maximum	469
7.4.8.8.6.4	StandardDev	470
7.4.8.8.7	TtError	470
7.4.8.8.7.1	Current	470
7.4.8.8.7.2	Average	471
7.4.8.8.7.3	Maximum	471
7.4.8.8.7.4	StandardDev	472
7.4.8.8.8	UePower	472
7.4.8.8.8.1	Current	472
7.4.8.8.8.2	Average	473
7.4.8.8.8.3	Maximum	473
7.4.8.8.8.4	StandardDev	474
7.4.8.8.9	Current	474
7.4.8.8.10	Average	475
7.4.8.8.11	Maximum	476
7.4.8.8.12	StandardDev	477
7.4.8.9	CdError	478
7.4.8.9.1	Dpcch	478
7.4.8.9.1.1	Current	478
7.4.8.9.1.2	Average	479
7.4.8.9.1.3	Maximum	479
7.4.8.9.1.4	StandardDev	480
7.4.8.9.2	Dpdch	480
7.4.8.9.2.1	Current	480
7.4.8.9.2.2	Average	481
7.4.8.9.2.3	Maximum	481
7.4.8.9.2.4	StandardDev	482
7.4.8.9.3	Hsdpcch	482
7.4.8.9.3.1	Current	482
7.4.8.9.3.2	Average	483
7.4.8.9.3.3	Maximum	483
7.4.8.9.3.4	StandardDev	484
7.4.8.9.4	Edpcch	484
7.4.8.9.4.1	Current	484
7.4.8.9.4.2	Average	485
7.4.8.9.4.3	Maximum	485
7.4.8.9.4.4	StandardDev	486
7.4.8.9.5	Edpdch<EdpdChannel>	486
7.4.8.9.5.1	Current	486
7.4.8.9.5.2	Average	487
7.4.8.9.5.3	Maximum	487
7.4.8.9.5.4	StandardDev	488
7.4.8.9.6	Current	488
7.4.8.9.7	Average	489
7.4.8.9.8	Maximum	490

		7.4.8.9.9	StandardDev	491	
7.5	Tpc			492	
	7.5.1	Dhib		495	
		7.5.1.1	Maximum	495	
		7.5.1.2	Minimum	496	
		7.5.1.3	Average	497	
		7.5.1.4	Statistics	498	
		7.5.1.5	Minimumc	499	
	7.5.2	Carrier<Carrier>		499	
		7.5.2.1	Psteps	500	
			7.5.2.1.1 Maximum	500	
			7.5.2.1.2 Minimum	503	
			7.5.2.1.3 Average	505	
			7.5.2.1.4 Statistics	508	
		7.5.2.2	UePower	509	
			7.5.2.2.1 Maximum	510	
			7.5.2.2.2 Minimum	511	
			7.5.2.2.3 Average	513	
			7.5.2.2.4 Statistics	514	
		7.5.2.3	Trace	515	
			7.5.2.3.1 UePower	515	
				7.5.2.3.1.1 Current	516
			7.5.2.3.2 Psteps	517	
				7.5.2.3.2.1 Current	517
	7.5.3	Total		519	
		7.5.3.1	UePower	519	
			7.5.3.1.1 Maximum	519	
			7.5.3.1.2 Minimum	520	
			7.5.3.1.3 Average	521	
			7.5.3.1.4 Statistics	521	
		7.5.3.2	Trace	522	
			7.5.3.2.1 UePower	522	
				7.5.3.2.1.1 Current	523
	7.5.4	State		524	
		7.5.4.1	All	524	
7.6	Prach			525	
	7.6.1	State		528	
		7.6.1.1	All	528	
	7.6.2	Trace		529	
		7.6.2.1	UePower	529	
			7.6.2.1.1 Current	530	
			7.6.2.1.2 Chip	530	
				7.6.2.1.2.1 Current	531
		7.6.2.2	EvMagnitude	531	
			7.6.2.2.1 Rms	532	
				7.6.2.2.1.1 Current	532
			7.6.2.2.2 Peak	533	
				7.6.2.2.2.1 Current	533
			7.6.2.2.3 Chip	534	
				7.6.2.2.3.1 Current	534
		7.6.2.3	Merror	535	
			7.6.2.3.1 Rms	535	
				7.6.2.3.1.1 Current	535
			7.6.2.3.2 Peak	536	

	7.6.2.3.2.1	Current	536
	7.6.2.3.3	Chip	537
	7.6.2.3.3.1	Current	537
7.6.2.4	Perror		538
	7.6.2.4.1	Rms	538
	7.6.2.4.1.1	Current	538
	7.6.2.4.2	Peak	539
	7.6.2.4.2.1	Current	539
	7.6.2.4.3	Chip	540
	7.6.2.4.3.1	Current	540
7.6.2.5	FreqError		541
	7.6.2.5.1	Current	541
7.6.2.6	Psteps		542
	7.6.2.6.1	Current	542
7.6.2.7	Iq		543
	7.6.2.7.1	Current	543
7.6.3	OffPower		544
7.6.4	Preamble<Preamble>		545
	7.6.4.1	Current	545
7.7	OoSync		547
	7.7.1	State	551
	7.7.1.1	All	552
7.8	OlpControl		552
	7.8.1	State	556
	7.8.1.1	All	557
	7.8.2	Carrier<CARRierExt>	557
	7.8.2.1	UepPower	558
	7.8.2.1.1	Rup<RampUpCarrier>	558



GETTING STARTED

1.1 Introduction



RsCmwWcdmaMeas 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

RsCmwWcdmaMeas 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 RsCmwWcdmaMeas`

Option 2 - Installing in Pycharm

- In Pycharm Menu **File->Settings->Project->Project Interpreter** click on the '+' button on the bottom left
- Type `RsCmwWcdmaMeas` 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 RsCmwWcdmaMeas offline:

- Download this python script (**Save target as**): `rsinstrument_offline_install.py` This installs all the preconditions that the RsCmwWcdmaMeas needs.
- Execute the script in your offline computer (supported is python 3.6 or newer)
- Download the RsCmwWcdmaMeas package to your computer from the pypi.org: <https://pypi.org/project/RsCmwWcdmaMeas/#files> to 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\RsCmwWcdmaMeas-3.8.10.5.tar`

1.3 Finding Available Instruments

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

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

from RsCmwWcdmaMeas import *

# Use the instr_list string items as resource names in the RsCmwWcdmaMeas constructor
instr_list = RsCmwWcdmaMeas.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 RsCmwWcdmaMeas 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 = RsCmwWcdmaMeas.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

RsCmwWcdmaMeas 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 RsCmwWcdmaMeas object. Below, is a simple Hello World example. Different resource names are examples for different physical interfaces.

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

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

from RsCmwWcdmaMeas import *

# A good practice is to assure that you have a certain minimum version installed
RsCmwWcdmaMeas.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 = RsCmwWcdmaMeas(resource_string_1)

idn = driver.utilities.query_str('*IDN?')
print(f"\nHello, I am: '{idn}'")
print(f'RsCmwWcdmaMeas 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; RsCmwWcdmaMeas 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 = RsCmwWcdmaMeas('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 `RsCmwWcdmaMeas` 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 `RsCmwWcdmaMeas` allows you to choose which VISA to use:

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

from RsCmwWcdmaMeas import *

# Force use of the Rs Visa. For NI Visa, use the "SelectVisa='ni'"
driver = RsCmwWcdmaMeas('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, `RsCmwWcdmaMeas` has something for you too - **no Visa installation raw LAN socket**:

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

from RsCmwWcdmaMeas import *

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

More option_string tokens are separated by comma:

```
driver = RsCmwWcdmaMeas('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 RsCmwWcdmaMeas objects:

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

from RsCmwWcdmaMeas import *

driver1 = RsCmwWcdmaMeas('TCPIP::192.168.56.101::INSTR', True, True)
driver2 = RsCmwWcdmaMeas.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 RsCmwWcdmaMeas 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 RsCmwWcdmaMeas import *

driver = RsCmwWcdmaMeas('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 RsCmwWcdmaMeas import *

driver = RsCmwWcdmaMeas('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 `RsCmwWcdmaMeas` raises an exception. Speaking of exceptions, an important feature of the `RsCmwWcdmaMeas` 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 RsCmwWcdmaMeas import *

driver = RsCmwWcdmaMeas('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

RsCmwWcdmaMeas 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 RsCmwWcdmaMeas 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 RsCmwWcdmaMeas 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 = RsCmwWcdmaMeas('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 RsCmwWcdmaMeas exceptions
    print(e.args[0])
    print('Some other RsCmwWcdmaMeas 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 RsCmwWcdmaMeas, 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 RsCmwWcdmaMeas 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 RsCmwWcdmaMeas has this covered. And, this feature is quite universal - not just for big files transfer, but for any data in both directions.

RsCmwWcdmaMeas 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 RsCmwWcdmaMeas 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 = RsCmwWcdmaMeas('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 `RsCmwWcdmaMeas` 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, `RsCmwWcdmaMeas` 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 RsCmwWcdmaMeas object
"""

import threading
from RsCmwWcdmaMeas import *
```

(continues on next page)

(continued from previous page)

```

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

driver = RsCmwWcdmaMeas('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 RsCmwWcdmaMeas objects with shared session
"""

import threading
from RsCmwWcdmaMeas import *

def execute(session: RsCmwWcdmaMeas, 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 = RsCmwWcdmaMeas('TCPIP::192.168.56.101::INSTR')
driver2 = RsCmwWcdmaMeas.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 RsCmwWcdmaMeas takes care of it for you. The text below describes this scenario.

Run the following example:

```

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

import threading
from RsCmwWcdmaMeas import *

def execute(session: RsCmwWcdmaMeas, 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 = RsCmwWcdmaMeas('TCPIP::192.168.56.101::INSTR')
driver2 = RsCmwWcdmaMeas('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 AclrMode

```
# Example value:  
value = enums.AclrMode.ABSolute  
# All values (2x):  
ABSolute | RELative
```

3.2 ActiveLimit

```
# Example value:  
value = enums.ActiveLimit.PC1  
# All values (6x):  
PC1 | PC2 | PC3 | PC3B | PC4 | USER
```

3.3 AnalysisMode

```
# Example value:  
value = enums.AnalysisMode.NOOFfset  
# All values (2x):  
NOOFfset | WOOFfset
```

3.4 AutoManualMode

```
# Example value:  
value = enums.AutoManualMode.AUTO  
# All values (2x):  
AUTO | MANual
```

3.5 Band

```
# First value:
value = enums.Band.OB1
# Last value:
value = enums.Band.OBS3
# All values (28x):
OB1 | OB10 | OB11 | OB12 | OB13 | OB14 | OB15 | OB16
OB17 | OB18 | OB19 | OB2 | OB20 | OB21 | OB22 | OB25
OB26 | OB3 | OB4 | OB5 | OB6 | OB7 | OB8 | OB9
OBL1 | OBS1 | OBS2 | OBS3
```

3.6 Carrier

```
# Example value:
value = enums.Carrier.C1
# All values (2x):
C1 | C2
```

3.7 CmwsConnector

```
# First value:
value = enums.CmwsConnector.R11
# Last value:
value = enums.CmwsConnector.RH8
# All values (96x):
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
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 | RF2 | RF3 | RF4 | RF5 | RF6 | RF7 | RF8
RG1 | RG2 | RG3 | RG4 | RG5 | RG6 | RG7 | RG8
RH1 | RH2 | RH3 | RH4 | RH5 | RH6 | RH7 | RH8
```


3.8 DetectionMode

```
# Example value:  
value = enums.DetectionMode.A3G  
# All values (1x):  
A3G
```

3.9 LimitHmode

```
# Example value:  
value = enums.LimitHmode.A  
# All values (3x):  
A | B | C
```

3.10 MeasMode

```
# Example value:  
value = enums.MeasMode.CTFC  
# All values (6x):  
CTFC | DHIB | ILPControl | MONitor | MPEDch | ULCM
```

3.11 MeasPeriod

```
# Example value:  
value = enums.MeasPeriod.FULLslot  
# All values (2x):  
FULLslot | HALFslot
```

3.12 Mode

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

3.13 Modulation

```
# Example value:  
value = enums.Modulation._4PAM  
# All values (5x):  
_4PAM | _4PVar | BPSK | BVAR | OFF
```

3.14 OutPowFstate

```
# Example value:  
value = enums.OutPowFstate.NOFF  
# All values (4x):  
NOFF | NON | OFF | ON
```

3.15 ParameterSetMode

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

3.16 PatternType

```
# Example value:  
value = enums.PatternType.AF  
# All values (3x):  
AF | AR | B
```

3.17 PcdErrorPhase

```
# Example value:  
value = enums.PcdErrorPhase.IPHase  
# All values (2x):  
IPHase | QPHase
```

3.18 Repeat

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

3.19 ResourceState

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

3.20 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
```

3.21 Retrigger

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

3.22 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
```

(continues on next page)

(continued from previous page)

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						

3.23 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
```

3.24 SetType

```
# First value:
value = enums.SetType.ALL0
# Last value:
value = enums.SetType.ULCM
# All values (19x):
ALL0 | ALL1 | ALternating | CLOop | CONTinuous | CTFC | DHIB | MPEDch
PHDown | PHUP | SAL0 | SAL1 | SALT | TSABc | TSE | TSEF
TSF | TSGH | ULCM
```

3.25 SignalSlope

```
# Example value:
value = enums.SignalSlope.FEDGE
# All values (2x):
FEDGE | REDGE
```

3.26 SlotNumber

```
# First value:
value = enums.SlotNumber.ANY
# Last value:
value = enums.SlotNumber.SL9
# All values (16x):
ANY | SL0 | SL1 | SL10 | SL11 | SL12 | SL13 | SL14
SL2 | SL3 | SL4 | SL5 | SL6 | SL7 | SL8 | SL9
```

3.27 SpreadingFactorA

```
# Example value:
value = enums.SpreadingFactorA.SF128
# All values (7x):
SF128 | SF16 | SF256 | SF32 | SF4 | SF64 | SF8
```

3.28 SpreadingFactorB

```
# First value:
value = enums.SpreadingFactorB._128
# Last value:
value = enums.SpreadingFactorB.V8
# All values (16x):
_128 | _16 | _2 | _256 | _32 | _4 | _64 | _8
V128 | V16 | V2 | V256 | V32 | V4 | V64 | V8
```

3.29 State

```
# Example value:
value = enums.State.OFF
# All values (3x):
OFF | ON | VAR
```

3.30 StopCondition

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

3.31 TestCase

```
# Example value:  
value = enums.TestCase.T0DB  
# All values (2x):  
T0DB | T1DB
```

3.32 TestScenarioB

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

3.33 Type

```
# Example value:  
value = enums.Type.ACK  
# All values (3x):  
ACK | CQI | NACK
```

3.34 UIConfiguration

```
# First value:  
value = enums.UIConfiguration._3CHS  
# Last value:  
value = enums.UIConfiguration.WCDMa  
# All values (16x):  
_3CHS | _3DUPlus | _3H DU | _4CHS | _4DUPlus | _4H DU | DCHS | DDUPlus  
DHDU | HDUPlus | HSDPa | HSPA | HSPLus | HSUPa | QPSK | WCDMa
```

REPCAPS

4.1 Instance (Global)

```
# Setting:
driver.repcap_instance_set(repcap.Instance.Inst1)
# Range:
Inst1 .. Inst32
# All values (32x):
Inst1 | Inst2 | Inst3 | Inst4 | Inst5 | Inst6 | Inst7 | Inst8
Inst9 | Inst10 | Inst11 | Inst12 | Inst13 | Inst14 | Inst15 | Inst16
Inst17 | Inst18 | Inst19 | Inst20 | Inst21 | Inst22 | Inst23 | Inst24
Inst25 | Inst26 | Inst27 | Inst28 | Inst29 | Inst30 | Inst31 | Inst32
```

4.2 Carrier

```
# First value:
value = repcap.Carrier.Nr1
# Values (2x):
Nr1 | Nr2
```

4.3 CARRierExt

```
# First value:
value = repcap.CARRierExt.Nr1
# Range:
Nr1 .. Nr32
# All values (32x):
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
```

4.4 EdpdChannel

```
# First value:  
value = repcap.EdpdChannel.Nr1  
# Values (4x):  
Nr1 | Nr2 | Nr3 | Nr4
```

4.5 Minus

```
# First value:  
value = repcap.Minus.Ch1  
# Values (2x):  
Ch1 | Ch2
```

4.6 Plus

```
# First value:  
value = repcap.Plus.Ch1  
# Values (2x):  
Ch1 | Ch2
```

4.7 Preamble

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

4.8 RampUpCarrier

```
# First value:  
value = repcap.RampUpCarrier.Nr1  
# Range:  
Nr1 .. Nr32  
# All values (32x):  
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
```


4.9 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
```


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()
```


RSCMWWCDMAMEAS API STRUCTURE

Global RepCaps

```
driver = RsCmwWcdmaMeas('TCPIP::192.168.2.101::HISLIP')
# Instance range: Inst1 .. Inst32
rc = driver.repcap_instance_get()
driver.repcap_instance_set(repcap.Instance.Inst1)
```

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

842 total commands, 8 Sub-groups, 0 group commands

Initializes new RsCmwWcdmaMeas 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 RsCmwWcdmaMeas session.

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

Creates a new RsCmwWcdmaMeas 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:WCDMA: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.TestScenarioB: SALone | CSPath | MAPRotocol SALone: ‘Standalone (Non Signaling)’ CSPath: ‘Combined Signal Path’ MAPRotocol: ‘Measure@Protocol Test’
- Controller: str: string Controlling application for scenario CSPath or MAPRotocol
- 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:WCDMA: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:WCDMA:MEASurement<Instance>:SCENario:SALone
ROUTE:WCDMA:MEASurement<Instance>:SCENario:CSPath
ROUTE:WCDMA: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:WCDMa: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 settings, analyzer settings and UE signal info settings while the combined signal path scenario is active.

return master: string String parameter selecting the master application For example, 'WCDMA Sig1' or 'WCDMA Sig2'

get_salone() → SaloneStruct

```
# SCPI: ROUTe:WCDMa: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() → RsCmwWcdmaMeas.enums.TestScenarioB

```
# SCPI: ROUTe:WCDMa:MEASurement<instance>:SCENario
value: enums.TestScenarioB = driver.route.scenario.get_value()
```

Returns the active scenario.

return scenario: SALone | CSPaTh | MAPRotocol SALone: 'Standalone (Non Signaling)' CSPaTh: 'Combined Signal Path' MAPRotocol: 'Measure@Protocol Test'

set_cspath(master: str) → None

```
# SCPI: ROUTe:WCDMa: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 settings, analyzer settings and UE signal info settings while the combined signal path scenario is active.

param master string String parameter selecting the master application For example, 'WCDMA Sig1' or 'WCDMA Sig2'

set_salone(value: RsCmwWcdmaMeas.Implementations.Route_.Scenario.Scenario.SaloneStruct) → None

```
# SCPI: ROUTe:WCDMa: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 MaProtocol

SCPI Commands

```
ROUTE:WCDMA:MEASurement<Instance>:SCENario:MAProtocol
```

class MaProtocol

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

set(controler: Optional[str] = None) → None

```
# SCPI: ROUTE:WCDMA:MEASurement<instance>:SCENario:MAProtocol
driver.route.scenario.maProtocol.set(controler = '1')
```

Activates the [Measure@ProtocolTest](#) scenario and optionally selects the controlling protocol test application. The signal routing and analyzer settings of the measurement application are ignored. Configure the corresponding settings within the protocol test application used in parallel.

param controler string String parameter selecting the protocol test application For example, 'Protocol Test1'

7.2 Configure

class Configure

Configure commands group definition. 161 total commands, 10 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.2.1 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.configure.carrier.repcap_carrier_get()
driver.configure.carrier.repcap_carrier_set(repcap.Carrier.Nr1)
```

class Carrier

Carrier commands group definition. 1 total commands, 1 Sub-groups, 0 group commands Repeated Capability: Carrier, default value after init: Carrier.Nr1

Cloning the Group

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

Subgroups**7.2.1.1 Band****SCPI Commands**

```
CONFIGure:WCDMa:MEASurement<Instance>:CARRier<Carrier>:BAND
```

class Band

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

get(*carrier*=<Carrier.Default: -1>) → RsCmwWcdmaMeas.enums.Band

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:CARRier<carrier>:BAND
value: enums.Band = driver.configure.carrier.band.get(carrier = repcap.Carrier.
↳Default)
```

Selects the operating band (OB) . INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:WCDMa:SIGN<i>:CARRier<c>:BAND
- CONFIGure:WCDMa:SIGN<i>:RFSettings:DBDC

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return band: OB1 | ... | OB14 | OB19 | ... | OB22 | OB25 | OB26 | OBS1 | ... | OBS3 | OBL1
 OB1, ..., OB14: operating band I to XIV OB19, ..., OB22: operating band XIX to XXII
 OB25, OB26: operating band XXV and XXVI OBS1: operating band S OBS2: operating band S 170 MHz
 OBS3: operating band S 190 MHz OBL1: operating band L
 Unit: OB1

set(*band*: RsCmwWcdmaMeas.enums.Band, *carrier*=<Carrier.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:CARRier<carrier>:BAND
driver.configure.carrier.band.set(band = enums.Band.OB1, carrier = repcap.
↳Carrier.Default)
```

Selects the operating band (OB) . INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:WCDMa:SIGN<i>:CARRier<c>:BAND
- CONFIGure:WCDMa:SIGN<i>:RFSettings:DBDC

param band OB1 | ... | OB14 | OB19 | ... | OB22 | OB25 | OB26 | OBS1 | ... | OBS3 | OBL1
 OB1, ..., OB14: operating band I to XIV OB19, ..., OB22: operating band XIX to XXII
 OB25, OB26: operating band XXV and XXVI OBS1: operating band S OBS2: operating band S 170 MHz
 OBS3: operating band S 190 MHz OBL1: operating band L
 Unit: OB1

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.2 Cell

class Cell

Cell 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.cell.clone()
```

Subgroups

7.2.2.1 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.configure.cell.carrier.repcap_carrier_get()
driver.configure.cell.carrier.repcap_carrier_set(repcap.Carrier.Nr1)
```

class Carrier

Carrier commands group definition. 1 total commands, 1 Sub-groups, 0 group commands Repeated Capability: Carrier, default value after init: Carrier.Nr1

Cloning the Group

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

Subgroups

7.2.2.1.1 Scode

SCPI Commands

```
CONFigure:WCDma:MEASurement<Instance>:CELL:CARRier<Carrier>:SCode
```

class SCode

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

get(*carrier*=<Carrier.Default: -1>) → float

```
# SCPI: CONFigure:WCDma:MEASurement<instance>:CELL:CARRier<carrier>:SCode
value: float = driver.configure.cell.carrier.scode.get(carrier = repcap.Carrier.
↳Default)
```

Specifies index i for calculation of the primary downlink scrambling code number by multiplication with 16. For the combined signal path scenario, useCONFigure:WCDma:SIGN<i>:CELL:CARRier<c>:SCode.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return code: numeric Range: #H0 to #H1FF

set(*code*: float, *carrier*=<Carrier.Default: -1>) → None

```
# SCPI: CONFigure:WCDma:MEASurement<instance>:CELL:CARRier<carrier>:SCode
driver.configure.cell.carrier.scode.set(code = 1.0, carrier = repcap.Carrier.
↳Default)
```

Specifies index i for calculation of the primary downlink scrambling code number by multiplication with 16. For the combined signal path scenario, useCONFigure:WCDma:SIGN<i>:CELL:CARRier<c>:SCode.

param code numeric Range: #H0 to #H1FF

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.3 UeSignal

SCPI Commands

```
CONFigure:WCDma:MEASurement<Instance>:UESignal:DPDCh
CONFigure:WCDma:MEASurement<Instance>:UESignal:ULConfig
CONFigure:WCDma:MEASurement<Instance>:UESignal:SFormat
CONFigure:WCDma:MEASurement<Instance>:UESignal:CMPattern
```

class UeSignal

UeSignal commands group definition. 5 total commands, 1 Sub-groups, 4 group commands

get_cm_pattern() → RsCmwWcdmaMeas.enums.PatternType

```
# SCPI: CONFigure:WCDma:MEASurement<instance>:UESignal:CMPattern
value: enums.PatternType = driver.configure.ueSignal.get_cm_pattern()
```

Selects the expected TPC pattern for UL compressed mode. INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:WCDMa:SIGN<i>:CMODE:ULCM:TYPE
- CONFIGure:WCDMa:SIGN<i>:CMODE:ULCM:ACTivation

return pattern_type: AR | AF | B AR: pattern A (rising TPC) defined in 3GPP TS 34.121, table 5.7.6 AF: pattern A (falling TPC) defined in 3GPP TS 34.121, table 5.7.7 B: pattern B defined in 3GPP TS 34.121, table 5.7.8

get_dpdch() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:DPDCh
value: bool = driver.configure.ueSignal.get_dpdch()
```

Defines whether the UL DPCH contains a DPDCH. For the combined signal path scenario, use CONFIGure:WCDMa:SIGN<i>:DL:LEVel:DPCH.

return dpdch: OFF | ON OFF: DPCCH only ON: DPCCH plus DPDCH

get_sformat() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:SFormat
value: int = driver.configure.ueSignal.get_sformat()
```

Selects the slot format for the UL DPCCH.

return slot_format: decimal Range: 0 to 5

get_ul_config() → RsCmwWcdmaMeas.enums.UlConfiguration

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:ULConfig
value: enums.UlConfiguration = driver.configure.ueSignal.get_ul_config()
```

Selects the uplink signal configuration.

return ul_configuration: QPSK | WCDMa | HSDPa | HSUPa | HSPA | HSPLus | DCHS | HDUPlus | DDUPlus | DHdu | 3CHS | 3DUPlus | 3HDU | 4CHS | 4DUPlus | 4HDU
 QPSK: QPSK signal WCDMa: WCDMA R99 signal HSDPa: signal with HSDPA-related channels HSUPa: signal with HSUPA channels HSPA: HSDPA related and HSUPA channels HSPLus: HSDPA+ related channels HDUPlus: HSDPA+ related and HSUPA channels DHdu: dual carrier HSDPA+ and dual carrier HSUPA active
 The following values cannot be set, but can be returned while the combined signal path scenario is active: DCHS: dual carrier HSDPA+ active DDUPlus: dual carrier HSDPA+ and HSUPA active 3CHS: three carrier HSDPA+ active 3DUPlus: three carrier HSDPA+ and HSUPA active 3HDU: three carrier HSDPA+ and dual carrier HSUPA active 4CHS: four carrier HSDPA+ active 4DUPlus: four carrier HSDPA+ and HSUPA active 4HDU: four carrier HSDPA+ and dual carrier HSUPA active

set_cm_pattern(pattern_type: RsCmwWcdmaMeas.enums.PatternType) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:CMPattern
driver.configure.ueSignal.set_cm_pattern(pattern_type = enums.PatternType.AF)
```

Selects the expected TPC pattern for UL compressed mode. INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFIGure:WCDMa:SIGN<i>:CMODE:ULCM:TYPE
- CONFIGure:WCDMa:SIGN<i>:CMODE:ULCM:ACTivation

param pattern_type AR | AF | B AR: pattern A (rising TPC) defined in 3GPP TS 34.121, table 5.7.6 AF: pattern A (falling TPC) defined in 3GPP TS 34.121, table 5.7.7 B: pattern B defined in 3GPP TS 34.121, table 5.7.8

set_dpdch(dpdch: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:DPDCh
driver.configure.ueSignal.set_dpdch(dpdch = False)
```

Defines whether the UL DPCH contains a DPDCH. For the combined signal path scenario, use CONFIGure:WCDMa:SIGN<i>:DL:LEVel:DPCH.

param dpdch OFF | ON OFF: DPCCH only ON: DPCCH plus DPDCH

set_sformat(slot_format: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:SFORMAT
driver.configure.ueSignal.set_sformat(slot_format = 1)
```

Selects the slot format for the UL DPCCH.

param slot_format decimal Range: 0 to 5

set_ul_config(ul_configuration: RsCmwWcdmaMeas.enums.UlConfiguration) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:ULConfig
driver.configure.ueSignal.set_ul_config(ul_configuration = enums.
    UlConfiguration._3CHS)
```

Selects the uplink signal configuration.

param ul_configuration QPSK | WCDMa | HSDPa | HSUPa | HSPA | HSPLus | DCHS | HDUPlus | DDUPlus | DHdu | 3CHS | 3DUPlus | 3Hdu | 4CHS | 4DUPlus | 4Hdu
 QPSK: QPSK signal WCDMa: WCDMA R99 signal HSDPa: signal with HSDPA-related channels HSUPa: signal with HSUPA channels HSPA: HSDPA related and HSUPA channels HSPLus: HSDPA+ related channels HDUPlus: HSDPA+ related and HSUPA channels DHdu: dual carrier HSDPA+ and dual carrier HSUPA active
 The following values cannot be set, but can be returned while the combined signal path scenario is active: DCHS: dual carrier HSDPA+ active DDUPlus: dual carrier HSDPA+ and HSUPA active 3CHS: three carrier HSDPA+ active 3DUPlus: three carrier HSDPA+ and HSUPA active 3Hdu: three carrier HSDPA+ and dual carrier HSUPA active 4CHS: four carrier HSDPA+ active 4DUPlus: four carrier HSDPA+ and HSUPA active 4Hdu: four carrier HSDPA+ and dual carrier HSUPA active

Cloning the Group

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

Subgroups

7.2.3.1 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.configure.ueSignal.carrier.repcap_carrier_get()
driver.configure.ueSignal.carrier.repcap_carrier_set(repcap.Carrier.Nr1)
```

class Carrier

Carrier commands group definition. 1 total commands, 1 Sub-groups, 0 group commands Repeated Capability: Carrier, default value after init: Carrier.Nr1

Cloning the Group

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

Subgroups

7.2.3.1.1 SCode

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:UESignal:CARRier<Carrier>:SCODE
```

class SCode

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

get(carrier=<Carrier.Default: -1>) → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UESignal:CARRier<carrier>:SCODE
value: float = driver.configure.ueSignal.carrier.scode.get(carrier = repcap.
↳Carrier.Default)
```

Selects the number of the long code that is used to scramble the received uplink WCDMA signal. For the combined signal path scenario, useCONFigure:WCDMa:SIGN<i>:UL:CARRier<c>:SCODE.

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

return code: numeric Range: #H0 to #HFFFFFF

set(code: float, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:CARRier<carrier>:SCODE
driver.configure.ueSignal.carrier.scode.set(code = 1.0, carrier = repcap.
↳Carrier.Default)
```

Selects the number of the long code that is used to scramble the received uplink WCDMA signal. For the combined signal path scenario, useCONFIGure:WCDMa:SIGN<i>:UL:CARRier<c>:SCODE.

param code numeric Range: #H0 to #FFFFFF

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.4 UeChannels

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:UEChannels:BSFSelection
```

class UeChannels

UeChannels commands group definition. 7 total commands, 1 Sub-groups, 1 group commands

get_bsf_selection() → RsCmwWcdmaMeas.enums.AutoManualMode

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UEChannels:BSFSelection
value: enums.AutoManualMode = driver.configure.ueChannels.get_bsf_selection()
```

Specifies the application controlling beta factor and spreading factor configuration in combined signal path.

return selection: AUTO | MANual AUTO: settings controlled by WCDMA signaling
MAN: settings controlled by WCDMA UE TX measurement

set_bsf_selection(selection: RsCmwWcdmaMeas.enums.AutoManualMode) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UEChannels:BSFSelection
driver.configure.ueChannels.set_bsf_selection(selection = enums.AutoManualMode.
↳AUTO)
```

Specifies the application controlling beta factor and spreading factor configuration in combined signal path.

param selection AUTO | MANual AUTO: settings controlled by WCDMA signaling
MAN: settings controlled by WCDMA UE TX measurement

Cloning the Group

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

Subgroups

7.2.4.1 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.configure.ueChannels.carrier.repcap_carrier_get()
driver.configure.ueChannels.carrier.repcap_carrier_set(repcap.Carrier.Nr1)
```

class Carrier

Carrier commands group definition. 6 total commands, 5 Sub-groups, 0 group commands Repeated Capability: Carrier, default value after init: Carrier.Nr1

Cloning the Group

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

Subgroups

7.2.4.1.1 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.configure.ueChannels.carrier.edpdch.repcap_edpdChannel_get()
driver.configure.ueChannels.carrier.edpdch.repcap_edpdChannel_set(repcap.EdpdChannel.Nr1)
```

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:UEChannels:CARRier<Carrier>:EDPDch<EdpdChannel>
```

class Edpdch

Edpdch commands group definition. 1 total commands, 0 Sub-groups, 1 group commands Repeated Capability: EdpdChannel, default value after init: EdpdChannel.Nr1

class EdpdchStruct

Structure for setting input parameters. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 5 to 5655
- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

get(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → EdpdchStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>:EDPDch
↳<nr>
value: EdpdchStruct = driver.configure.ueChannels.carrier.edpdch.get(carrier =
↳repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.Default)
```

Specifies the presence of a selected E-DPDCH (1 to 4) in the uplink signal and the beta factor and spreading factor of the channel.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting: CONFIGure:WCDMa:SIGN<i>:UL:GFACtor:HSUPa:EDPCch
- Setting of spreading factor via automatic configuration depending on connection configuration

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

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

set(*structure*:

RsCmwWcdmaMeas.Implementations.Configure_.UeChannels_.Carrier_.Edpdch.Edpdch.EdpdchStruct,
carrier=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>:EDPDch
↳<nr>
driver.configure.ueChannels.carrier.edpdch.set(value = [PROPERTY_STRUCT_NAME](),
↳ carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.Default)
```

Specifies the presence of a selected E-DPDCH (1 to 4) in the uplink signal and the beta factor and spreading factor of the channel.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting: CONFIGure:WCDMa:SIGN<i>:UL:GFACtor:HSUPa:EDPCch
- Setting of spreading factor via automatic configuration depending on connection configuration

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

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

Cloning the Group

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

7.2.4.1.2 Edpcch

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:UEChannels:CARRier<Carrier>:EDPCch
```

class Edpcch

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

class EdpcchStruct

Structure for setting input parameters. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 5 to 3585
- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

get(carrier=<Carrier.Default: -1>) → EdpcchStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>:EDPCch
value: EdpcchStruct = driver.configure.ueChannels.carrier.edpcch.get(carrier =
↳ repcap.Carrier.Default)
```

Specifies the presence of an E-DPCCH in the uplink signal and the beta factor and spreading factor of the channel.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting: CONFigure:WCDMa:SIGN<i>:UL:GFACtor:HSUPa:EDPCch
- Setting of spreading factor via automatic configuration depending on connection configuration

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

set(structure:

RsCmwWcdmaMeas.Implementations.Configure_.UeChannels_.Carrier_.Edpcch.Edpcch.EdpcchStruct,
carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>:EDPCch
driver.configure.ueChannels.carrier.edpcch.set(value = [PROPERTY_STRUCT_NAME](),
↳ carrier = repcap.Carrier.Default)
```

Specifies the presence of an E-DPCCH in the uplink signal and the beta factor and spreading factor of the channel.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting: CONFigure:WCDMa:SIGN<i>:UL:GFACtor:HSUPa:EDPCch
- Setting of spreading factor via automatic configuration depending on connection configuration

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

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.4.1.3 Hsdpcch

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:UEChannels:CARRier<Carrier>:HSDPcch
```

class Hsdpcch

Hsdpcch commands group definition. 2 total commands, 1 Sub-groups, 1 group commands

class HsdpcchStruct

Structure for setting input parameters. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 5 to 570
- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

get(carrier=<Carrier.Default: -1>) → HsdpcchStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>
↳:HSDPcch
value: HsdpcchStruct = driver.configure.ueChannels.carrier.hsdpcch.get(carrier,
↳= repcap.Carrier.Default)
```

Specifies the presence of an HS-DPCCH in the uplink signal and the beta factor and spreading factor of the channel. For the HS-DPCCH three sets of beta factor and spreading factor can be configured, depending on whether it transports an ACK, NACK or CQI. This command configures/returns the values related to the currently active set. For selection of the active set, see method RsCmwWcdmaMeas.Configure.UeChannels.Carrier.Hsdpcch.Config.set.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting: CONFigure:WCDMa:SIGN<i>:UL:GFACtor:HSDPa
- Setting of spreading factor via automatic configuration depending on connection configuration

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

set(structure: RsCmwWcdmaMeas.Implementations.Configure_.UeChannels_.Carrier_.Hsdpcch.Hsdpcch.HsdpcchStruct, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>
↳:HSDPcch
driver.configure.ueChannels.carrier.hsdpcch.set(value = [PROPERTY_STRUCT_
↳NAME](), carrier = repcap.Carrier.Default)
```

Specifies the presence of an HS-DPCCH in the uplink signal and the beta factor and spreading factor of the channel. For the HS-DPCCH three sets of beta factor and spreading factor can be configured, depending on whether it transports an ACK, NACK or CQI. This command configures/returns the values related to the currently active set. For selection of the active set, see method RsCmwWcdmaMeas.Configure.UeChannels.Carrier.Hsdpcch.Config.set.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting: CONFigure:WCDMa:SIGN<i>:UL:GFACtor:HSDPa
- Setting of spreading factor via automatic configuration depending on connection configuration

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

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

Cloning the Group

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

Subgroups

7.2.4.1.3.1 Config

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:UEChannels:CARRier<Carrier>:HSDPcch:CONFig
```

class Config

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

get(carrier=<Carrier.Default: -1>) → RsCmwWcdmaMeas.enums.Type

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>
↳:HSDPcch:CONFig
value: enums.Type = driver.configure.ueChannels.carrier.hsdpcch.config.
↳get(carrier = repcap.Carrier.Default)
```

Selects whether the HS-DPCCH transports an ACK, NACK or CQI and thus which set of beta factor and spreading factor values is used.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting: CONFigure:WCDMa:SIGN<i>:UL:GFACtor:HSDPa
- Setting of spreading factor via automatic configuration depending on connection configuration

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return type_py: ACK | NACK | CQI

set(type_py: RsCmwWcdmaMeas.enums.Type, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>
↳:HSDPcch:CONFig
driver.configure.ueChannels.carrier.hsdpcch.config.set(type_py = enums.Type.ACK,
↳ carrier = repcap.Carrier.Default)
```

Selects whether the HS-DPCCH transports an ACK, NACK or CQI and thus which set of beta factor and spreading factor values is used.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting: CONFigure:WCDMa:SIGN<i>:UL:GFACtor:HSDPa
- Setting of spreading factor via automatic configuration depending on connection configuration

param type_py ACK | NACK | CQI

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.4.1.4 Dpdch

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:UECHannels:CARRier<Carrier>:DPDCh
```

class Dpdch

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

class DpdchStruct

Structure for setting input parameters. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 0 to 15
- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

get(carrier=<Carrier.Default: -1>) → DpdchStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>:DPDCh
value: DpdchStruct = driver.configure.ueChannels.carrier.dpdch.get(carrier =
↳ repcap.Carrier.Default)
```

Specifies the presence of a DPDCH in the uplink signal and the beta factor and spreading factor of the channel.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting:

Table Header:

- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:PDATa<no>
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:RMC<no>

- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:VIDeo
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:VOICE
- Setting of spreading factor via automatic configuration depending on connection

configuration

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

set(structure:

RsCmwWcdmaMeas.Implementations.Configure_.UeChannels_.Carrier_.Dpdch.Dpdch.DpdchStruct, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>:DPDCh
driver.configure.ueChannels.carrier.dpdch.set(value = [PROPERTY_STRUCT_NAME](),
↪carrier = repcap.Carrier.Default)
```

Specifies the presence of a DPDCH in the uplink signal and the beta factor and spreading factor of the channel.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting:

Table Header:

- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:PDATa<no>
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:RMC<no>
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:VIDeo
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:VOICE
- Setting of spreading factor via automatic configuration depending on connection

configuration

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

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.4.1.5 Dpcch

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:UECHannels:CARRier<Carrier>:DPCCh
```

class Dpcch

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

class DpcchStruct

Structure for setting input parameters. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 1 to 15

- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

get(*carrier*=<Carrier.Default: -1>) → DpcchStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>:DPCCh
value: DpcchStruct = driver.configure.ueChannels.carrier.dpcch.get(carrier =
↳repcap.Carrier.Default)
```

Specifies the presence of a DPCCH in the uplink signal and the beta factor and spreading factor of the channel.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting:

Table Header:

- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:PDATA<no>
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:RMC<no>
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:VIDeo
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:VOICe
- Setting of spreading factor via automatic configuration depending on connection

configuration

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

set(*structure*:

RsCmwWcdmaMeas.Implementations.Configure_UeChannels_Carrier_Dpcch.Dpcch.DpcchStruct,
carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>:DPCCh
driver.configure.ueChannels.carrier.dpcch.set(value = [PROPERTY_STRUCT_NAME](),
↳carrier = repcap.Carrier.Default)
```

Specifies the presence of a DPCCH in the uplink signal and the beta factor and spreading factor of the channel.

INTRO_CMD_HELP: For the combined signal path scenario, use:

- Beta factor setting:

Table Header:

- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:PDATA<no>
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:RMC<no>
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:VIDeo
- CONFigure:WCDMa:SIGN<i>:UL:GFACtor:VOICe
- Setting of spreading factor via automatic configuration depending on connection

configuration

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

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.5 RfSettings

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:RFSettings:EATTenuation
CONFigure:WCDMa:MEASurement<Instance>:RFSettings:UMARgin
CONFigure:WCDMa:MEASurement<Instance>:RFSettings:ENPower
```

class RfSettings

RfSettings commands group definition. 5 total commands, 2 Sub-groups, 3 group commands

get_eattenuation() → float

```
# SCPI: CONFigure:WCDMa: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:WCDMa:SIGN<i>:RFSettings:CARRIER<c>:EATTenuation:INPut.

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

get_envelope_power() → float

```
# SCPI: CONFigure:WCDMa: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:WCDMa:SIGN<i>:RFSettings:ENPMode
- CONFigure:WCDMa:SIGN<i>:RFSettings:ENPower

return exp_nom_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_umargin() → float

```
# SCPI: CONFigure:WCDMa: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. For the combined signal path scenario, useCONFigure:WCDMa:SIGN<i>:RFSettings:MARGIN.

return user_margin: numeric Range: 0 dB to (55 dB + external attenuation - expected nominal power) , Unit: dB

set_eattenuation(rf_input_ext_att: float) → None

```
# SCPI: CONFIGure:WCDMa: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, use CONFIGure:WCDMa:SIGN<i>:RFSettings:CARRier<c>:EATTenuation:INPut.

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

set_envelope_power(exp_nom_power: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:RFSettings:ENPower
driver.configure.rfSettings.set_envelope_power(exp_nom_power = 1.0)
```

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

- CONFIGure:WCDMa:SIGN<i>:RFSettings:ENPMode
- CONFIGure:WCDMa:SIGN<i>:RFSettings:ENPower

param exp_nom_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_umargin(user_margin: float) → None

```
# SCPI: CONFIGure:WCDMa: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. For the combined signal path scenario, use CONFIGure:WCDMa:SIGN<i>:RFSettings:MARGin.

param user_margin numeric Range: 0 dB to (55 dB + external attenuation - expected nominal power) , Unit: dB

Cloning the Group

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

Subgroups

7.2.5.1 Dcarrier

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:RFSettings:DCARrier:SEParation
```

class Dcarrier

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

get_separation() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:RFSettings:DCARrier:SEParation
value: float = driver.configure.rfSettings.dcarrier.get_separation()
```

Sets the carrier 1 and carrier 2 frequency separation for measurements with dual uplink carrier. For the combined signal path scenario, use CONFigure:WCDMa:SIGN<i>:RFSettings:DCARrier:SEParation.

return dc_freq_sep: numeric Range: 0 MHz to 10 MHz , Unit: Hz

set_separation(dc_freq_sep: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:RFSettings:DCARrier:SEParation
driver.configure.rfSettings.dcarrier.set_separation(dc_freq_sep = 1.0)
```

Sets the carrier 1 and carrier 2 frequency separation for measurements with dual uplink carrier. For the combined signal path scenario, use CONFigure:WCDMa:SIGN<i>:RFSettings:DCARrier:SEParation.

param dc_freq_sep numeric Range: 0 MHz to 10 MHz , Unit: Hz

7.2.5.2 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.configure.rfSettings.carrier.repcap_carrier_get()
driver.configure.rfSettings.carrier.repcap_carrier_set(repcap.Carrier.Nr1)
```

class Carrier

Carrier commands group definition. 1 total commands, 1 Sub-groups, 0 group commands Repeated Capability: Carrier, default value after init: Carrier.Nr1

Cloning the Group

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

Subgroups

7.2.5.2.1 Frequency

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:RFSettings:CARRier<Carrier>:FREQuency
```

class Frequency

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

get(carrier=<Carrier.Default: -1>) → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:RFSettings:CARRier<carrier>
↳:FREQuency
value: float = driver.configure.rfSettings.carrier.frequency.get(carrier =
↳repcap.Carrier.Default)
```

Selects the center frequency of the RF analyzer. INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFigure:WCDMa:SIGN<i>:RFSettings:CARRier<c>:FREQuency:UL
- CONFigure:WCDMa:SIGN<i>:RFSettings:CARRier<c>:FOFFset:UL
- CONFigure:WCDMa:SIGN<i>:RFSettings:CARRier<c>:CHANnel:UL

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 carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return frequency: numeric Range: 70 MHz to 6 GHz, Unit: Hz Using the unit CH the frequency can be set via the channel number. The allowed channel number range depends on the operating band, see ‘Operating Bands’.

set(frequency: float, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:RFSettings:CARRier<carrier>
↳:FREQuency
driver.configure.rfSettings.carrier.frequency.set(frequency = 1.0, carrier =
↳repcap.Carrier.Default)
```

Selects the center frequency of the RF analyzer. INTRO_CMD_HELP: For the combined signal path scenario, use:

- CONFigure:WCDMa:SIGN<i>:RFSettings:CARRier<c>:FREQuency:UL

- CONFIGure:WCDMa:SIGN<i>:RFSettings:CARRier<c>:FOFFset:UL
- CONFIGure:WCDMa:SIGN<i>:RFSettings:CARRier<c>:CHANnel:UL

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: 70 MHz to 6 GHz, Unit: Hz Using the unit CH the frequency can be set via the channel number. The allowed channel number range depends on the operating band, see ‘Operating Bands’.

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

7.2.6 MultiEval

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:TOUT
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:MSCount
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:PSLot
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:SYNch
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:MOEXception
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:SCONdition
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:REPetition
```

class MultiEval

MultiEval commands group definition. 72 total commands, 11 Sub-groups, 7 group commands

get_mo_exception() → bool

```
# SCPI: CONFIGure:WCDMa: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 OFF: Faulty results are rejected. ON: Results are never rejected.

get_ms_count() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:MSCount
value: int = driver.configure.multiEval.get_ms_count()
```

Selects the total number of measured slots.

return slot_count: decimal Range: 1 slot to 120 slots

get_pslot() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:PSLot
value: int = driver.configure.multiEval.get_pslot()
```

Selects the slot where the R&S CMW calculates the results of single slot measurements: ACLR, emission mask, EVM vs. chip, CD monitor. The number of the preselected slot must be smaller than the number of measured slots (method RsCmwWcdmaMeas. Configure.MultiEval.msCount) .

return slot_number: integer Range: 0 to 119

get_repetition() → RsCmwWcdmaMeas.enums.Repeat

```
# SCPI: CONFIGure:WCDMa: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>:::SCOunt to determine the number of measurement intervals per single shot.

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

get_scondition() → RsCmwWcdmaMeas.enums.StopCondition

```
# SCPI: CONFIGure:WCDMa: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 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_synch() → RsCmwWcdmaMeas.enums.SlotNumber

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SYNCh
value: enums.SlotNumber = driver.configure.multiEval.get_synch()
```

Selects a slot number within the UL WCDMA frames (0 to 14) that the R&S CMW displays as the first slot in the measurement interval.

return slot_number: ANY | SL1 | SL2 | SL3 | SL4 | SL5 | SL6 | SL7 | SL8 | SL9 | SL10 |
SL11 | SL12 | SL13 | SL14 | SL0
ANY: No frame synchronization
SL0 ... SL14: First slot = slot 0 ... slot 14

get_timeout() → float

```
# SCPI: CONFIGure:WCDMa: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_mo_exception(*meas_on_exception: bool*) → None

```
# SCPI: CONFIGure:WCDMa: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 OFF: Faulty results are rejected. ON: Results are never rejected.

set_ms_count(*slot_count: int*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:MSCount
driver.configure.multiEval.set_ms_count(slot_count = 1)
```

Selects the total number of measured slots.

param slot_count decimal Range: 1 slot to 120 slots

set_pslot(*slot_number: int*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:PSlot
driver.configure.multiEval.set_pslot(slot_number = 1)
```

Selects the slot where the R&S CMW calculates the results of single slot measurements: ACLR, emission mask, EVM vs. chip, CD monitor. The number of the preselected slot must be smaller than the number of measured slots (method RsCmwWcdmaMeas. Configure.MultiEval.msCount) .

param slot_number integer Range: 0 to 119

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

```
# SCPI: CONFIGure:WCDMa: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: RsCmwWcdmaMeas.enums.StopCondition*) → None

```
# SCPI: CONFIGure:WCDMa: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 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_synch(slot_number: RsCmwWcdmaMeas.enums.SlotNumber) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SYNCh
driver.configure.multiEval.set_synch(slot_number = enums.SlotNumber.ANY)
```

Selects a slot number within the UL WCDMA frames (0 to 14) that the R&S CMW displays as the first slot in the measurement interval.

param slot_number ANY | SL1 | SL2 | SL3 | SL4 | SL5 | SL6 | SL7 | SL8 | SL9 | SL10 |
SL11 | SL12 | SL13 | SL14 | SL0 ANY: No frame synchronization SL0 ... SL14: First
slot = slot 0 ... slot 14

set_timeout(timeout: float) → None

```
# SCPI: CONFIGure:WCDMa: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

7.2.6.1 Scout

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:SCount:BER
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:SCount:MODulation
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:SCount:SPECtrum
```

class Scout

Scout commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

get_ber() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCount:BER
value: int = driver.configure.multiEval.scout.get_ber()
```

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

return statistic_count: decimal Number of transport blocks Range: 1 to 1000

get_modulation() → int

```
# SCPI: CONFIGure:WCDMa: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 statistic_count: decimal Number of measurement intervals Range: 1 to 1000

get_spectrum() → int

```
# SCPI: CONFIGure:WCDMa: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 statistic_count: decimal Number of measurement intervals Range: 1 to 1000

set_ber(statistic_count: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCount:BER
driver.configure.multiEval.scount.set_ber(statistic_count = 1)
```

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

param statistic_count decimal Number of transport blocks Range: 1 to 1000

set_modulation(statistic_count: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCount:MODulation
driver.configure.multiEval.scount.set_modulation(statistic_count = 1)
```

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

param statistic_count decimal Number of measurement intervals Range: 1 to 1000

set_spectrum(statistic_count: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCount:SPECtrum
driver.configure.multiEval.scount.set_spectrum(statistic_count = 1)
```

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

param statistic_count decimal Number of measurement intervals Range: 1 to 1000

7.2.6.2 Limit

SCPI Commands

```

CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:PHSDpcch
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:PHD
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:VMagnitude
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:MERRor
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:PERRor
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:IQOffset
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:IQIMbalance
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:CFError

```

class Limit

Limit commands group definition. 23 total commands, 4 Sub-groups, 8 group commands

class EvMagnitudeStruct

Structure for reading output parameters. Fields:

- Rms: float or bool: numeric | ON | OFF Range: 0 % to 100 %, Unit: % Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Peak: float or bool: numeric | ON | OFF Range: 0 % to 99 %, Unit: % Additional OFF | ON disables/enables the limit check using the previous/default limit values

class MerrorStruct

Structure for reading output parameters. Fields:

- Rms: float or bool: numeric | ON | OFF Range: 0 % to 100 %, Unit: % Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Peak: float or bool: numeric | ON | OFF Range: 0 % to 99 %, Unit: % Additional OFF | ON disables/enables the limit check using the previous/default limit values

class PerrorStruct

Structure for reading output parameters. Fields:

- Rms: float or bool: numeric | ON | OFF Range: 0 deg to 45 deg, Unit: deg Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Peak: float or bool: numeric | ON | OFF Range: 0 deg to 45 deg, Unit: deg Additional OFF | ON disables/enables the limit check using the previous/default limit values

class PhdStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Upper: float: numeric Range: 0 deg to 90 deg, Unit: deg
- Dynamic: float: numeric Range: 0 deg to 90 deg, Unit: deg

class PhsDpcchStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Measure_Point_A: float: numeric Range: 0.5 slots to 119.5 slots, Unit: slot
- Measure_Point_B: float: numeric Range: 0.5 slots to 119.5 slots, Unit: slot
- Dynamic: float: numeric Range: 0 deg to 90 deg, Unit: deg

get_cf_error() → float

```
# SCPI: CONFIGure:WCDMa: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 frequency_error: numeric | ON | OFF Range: 0 Hz to 4000 Hz, Unit: Hz Additional OFF | ON disables/enables the limit check using the previous/default limit values

get_ev_magnitude() → EvMagnitudeStruct

```
# SCPI: CONFIGure:WCDMa: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:WCDMa: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: numeric | ON | OFF Range: -99 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values

get_iq_offset() → float

```
# SCPI: CONFIGure:WCDMa: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.

return iq_offset: numeric | ON | OFF Range: -80 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values

get_merror() → MerrorStruct

```
# SCPI: CONFIGure:WCDMa: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_perror() → PerrorStruct

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

Defines symmetric limits 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 values.

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

get_phd() → PhdStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PHD
value: PhdStruct = driver.configure.multiEval.limit.get_phd()
```

Defines upper and dynamic limits for the phase discontinuity determined by full-slot measurements (signals without HSPA channels).

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

get_phs_dpcch() → PhsDpcchStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PHSDpcch
value: PhsDpcchStruct = driver.configure.multiEval.limit.get_phs_dpcch()
```

Defines a dynamic limit for the phase discontinuity determined by half-slot measurements (signals with HS-DPCCH). The limit is checked at point A and point B. As the phase discontinuity is measured at half-slot boundaries (x.5, not x. 0) points A and B have to be set to half-slot positions.

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

set_cf_error(frequency_error: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:CFError
driver.configure.multiEval.limit.set_cf_error(frequency_error = 1.0)
```

Defines an upper limit for the carrier frequency error.

param frequency_error numeric | ON | OFF Range: 0 Hz to 4000 Hz, Unit: Hz Additional OFF | ON disables/enables the limit check using the previous/default limit values

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

```
# SCPI: CONFIGure:WCDMa: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:WCDMa: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 numeric | ON | OFF Range: -99 dB to 0 dB, Unit: dB Additional
OFF | ON disables/enables the limit check using the previous/default limit values

set_iq_offset(*iq_offset: float*) → None

```
# SCPI: CONFIGure:WCDMa: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.

param iq_offset numeric | ON | OFF Range: -80 dB to 0 dB, Unit: dB Additional OFF
| ON disables/enables the limit check using the previous/default limit values

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

```
# SCPI: CONFIGure:WCDMa: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_perror(*value: RsCmwWcdmaMeas.Implementations.Configure_MultiEval_Limit.Limit.PerrorStruct*)
→ None

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

Defines symmetric limits 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 values.

param value see the help for PerrorStruct structure arguments.

set_phd(*value: RsCmwWcdmaMeas.Implementations.Configure_MultiEval_Limit.Limit.PhdStruct*) →
None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PHD
driver.configure.multiEval.limit.set_phd(value = PhdStruct())
```

Defines upper and dynamic limits for the phase discontinuity determined by full-slot measurements (signals without HSPA channels).

param value see the help for PhdStruct structure arguments.

set_phs_dpcch(*value: RsCmwWcdmaMeas.Implementations.Configure_MultiEval_Limit.Limit.PhsDpcchStruct*)
→ None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PHSDpcch
driver.configure.multiEval.limit.set_phs_dpcch(value = PhsDpcchStruct())
```

Defines a dynamic limit for the phase discontinuity determined by half-slot measurements (signals with HS-DPCCH). The limit is checked at point A and point B. As the phase discontinuity is measured at half-slot boundaries (x.5, not x. 0) points A and B have to be set to half-slot positions.

param value see the help for PhsDpcchStruct structure arguments.

Cloning the Group

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

Subgroups

7.2.6.2.1 RcdError

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:RCDerror:ECDP
```

class RcdError

RcdError commands group definition. 8 total commands, 1 Sub-groups, 1 group commands

class EcdpStruct

Structure for reading output parameters. Fields:

- Threshold_Bpsk_1: float: numeric Lower ECDP threshold for BPSK requirement 1 Range: -50 dB to 0 dB, Unit: dB
- Threshold_Bpsk_2: float: numeric Lower ECDP threshold for BPSK requirement 2 Range: -50 dB to 0 dB, Unit: dB
- Limit_Bpsk_1: float: numeric RCDE limit for BPSK requirement 1 Range: -50 dB to 0 dB, Unit: dB
- Limit_Bpsk_2: float: numeric RCDE limit for BPSK requirement 2 (limit = this value minus ECDP) Range: -50 dB to 0 dB, Unit: dB
- Threshold_4_Pam_1: float: numeric Lower ECDP threshold for 4PAM requirement 1 Range: -50 dB to 0 dB, Unit: dB
- Threshold_4_Pam_2: float: numeric Lower ECDP threshold for 4PAM requirement 2 Range: -50 dB to 0 dB, Unit: dB
- Limit_4_Pam_1: float: numeric RCDE limit for 4PAM requirement 1 Range: -50 dB to 0 dB, Unit: dB
- Limit_4_Pam_2: float: numeric RCDE limit for 4PAM requirement 2 (limit = this value minus ECDP) Range: -50 dB to 0 dB, Unit: dB

get_ecdp() → EcdpStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:RCDerror:ECDP
value: EcdpStruct = driver.configure.multiEval.limit.rcdError.get_ecdp()
```

Defines upper limits for the relative CDE (RCDE) of BPSK and 4PAM modulated channels. For each modulation type, two requirements are defined.

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

set_ecdp(value: RsCmwWcdmaMeas.Implementations.Configure_MultiEval_Limit_RcdError.RcdError.EcdpStruct) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIMit:RCDerror:ECDP
driver.configure.multiEval.limit.rcdError.set_ecdp(value = EcdpStruct())
```

Defines upper limits for the relative CDE (RCDE) of BPSK and 4PAM modulated channels. For each modulation type, two requirements are defined.

param value see the help for EcdpStruct structure arguments.

Cloning the Group

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

Subgroups

7.2.6.2.1.1 Eecdp

class Eecdp

Eecdp commands group definition. 7 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.limit.rcdError.eecdp.clone()
```

Subgroups

7.2.6.2.1.2 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.configure.multiEval.limit.rcdError.eecdp.carrier.repcap_carrier_get()
driver.configure.multiEval.limit.rcdError.eecdp.carrier.repcap_carrier_set(repcap.
↳ Carrier.Nr1)
```

SCPI Commands

`CONFigure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:RCDerror:EECDp:CARRier<Carrier>`

class Carrier

Carrier commands group definition. 7 total commands, 5 Sub-groups, 1 group commands Repeated Capability: Carrier, default value after init: Carrier.Nr1

class GetStruct

Response structure. Fields:

- Enable_Dpcch: bool: No parameter help available
- Beta_Dpcch: int: No parameter help available
- Sf_Dpcch: int: No parameter help available
- Nom_Cdp_Dpcch: float: No parameter help available
- Eff_Cdp_Dpcch: float: No parameter help available
- Enable_Dpdch: bool: No parameter help available
- Beta_Dpdch: int: No parameter help available
- Sf_Dpdch: int: No parameter help available
- Nom_Cdp_Dpdch: float: No parameter help available
- Eff_Cdp_Dpdch: float: No parameter help available
- Enable_Hs_Dpcch: bool: No parameter help available
- Beta_Hsdpch: int: No parameter help available
- Sf_Hs_Dpcch: int: No parameter help available
- Nom_Hs_Dpcch: float: No parameter help available
- Eff_Hs_Dpcch: float: No parameter help available
- Enable_Edpcch: bool: No parameter help available
- Beta_Edpcch: int: No parameter help available
- Sfe_Dpcch: int: No parameter help available
- Nom_Edpcch: float: No parameter help available
- Effe_Dpcch: float: No parameter help available
- Enable_Edpdch_1: bool: No parameter help available
- Beta_Edpdch_1: int: No parameter help available
- Sfe_Dpd_Ch_1: int: No parameter help available
- Nom_Edpdch_1: float: No parameter help available
- Eff_Edpdch_1: float: No parameter help available
- Enable_Edpdch_2: bool: No parameter help available
- Beta_Edpdch_2: int: No parameter help available
- Sfe_Dpd_Ch_2: int: No parameter help available
- Nom_Edpdch_2: float: No parameter help available

- Eff_Edpdch_2: float: No parameter help available
- Enable_Edpdch_3: bool: No parameter help available
- Beta_Edpdch_3: int: No parameter help available
- Sfe_Dpd_Ch_3: int: No parameter help available
- Nom_Edpdch_3: float: No parameter help available
- Eff_Edpdch_3: float: No parameter help available
- Enable_Edpdch_4: bool: No parameter help available
- Beta_Edpdch_4: int: No parameter help available
- Sfe_Dpd_Ch_4: int: No parameter help available
- Nom_Edpdch_4: float: No parameter help available
- Eff_Edpdch_4: float: No parameter help available

class SetStruct

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

- Enable_Dpcch: bool: No parameter help available
- Beta_Dpcch: int: No parameter help available
- Sf_Dpcch: int: No parameter help available
- Enable_Dpdch: bool: No parameter help available
- Beta_Dpdch: int: No parameter help available
- Sf_Dpdch: int: No parameter help available
- Enable_Hs_Dpcch: bool: No parameter help available
- Beta_Hsdpch: int: No parameter help available
- Sf_Hs_Dpcch: int: No parameter help available
- Enable_Edpcch: bool: No parameter help available
- Beta_Edpcch: int: No parameter help available
- Sfe_Dpcch: int: No parameter help available
- Enable_Edpdch_1: bool: No parameter help available
- Beta_Edpdch_1: int: No parameter help available
- Sfe_Dpd_Ch_1: int: No parameter help available
- Enable_Edpdch_2: bool: No parameter help available
- Beta_Edpdch_2: int: No parameter help available
- Sfe_Dpd_Ch_2: int: No parameter help available
- Enable_Edpdch_3: bool: No parameter help available
- Beta_Edpdch_3: int: No parameter help available
- Sfe_Dpd_Ch_3: int: No parameter help available
- Enable_Edpdch_4: bool: No parameter help available
- Beta_Edpdch_4: int: No parameter help available

- Sfe_Dpd_Ch_4: int: No parameter help available

get(carrier=<Carrier.Default: -1>) → GetStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>
value: GetStruct = driver.configure.multiEval.limit.rcdError.eecdP.carrier.
↳get(carrier = repcap.Carrier.Default)

INTRO_CMD_HELP: Specifies the channel configuration in the uplink signal.↳
↳This command has the same effect as the sum of the following commands:

- method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.
↳Dpcch.set
- method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.
↳Dpdch.set
- method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.
↳Hsdpcch.set
- method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.
↳Edpcch.set
- CONFIGure:WCDMa:MEAS<i>:MEvaluation:LIMit:RCDError:EECDp:CARRier<c>:EDPDch
↳<no>
```

Please refer to these commands for additional information (ranges and *RST values) . The parameter array described below is repeated for each channel (eight times) in the following order: DPCCH, DPDCH, HS-DPCCH, E-DPCCH, E-DPDCH 1, ... , E-DPDCH 4. Thus a setting requires 3*8 values and a query returns 5*8 values.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

set(structure: RsCmwWcd-

maMeas.Implementations.Configure_.MultiEval_.Limit_.RcdError_.EecdP_.Carrier.Carrier.SetStruct, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>
driver.configure.multiEval.limit.rcdError.eecdP.carrier.set(value = [PROPERTY_
↳STRUCT_NAME](), carrier = repcap.Carrier.Default)

INTRO_CMD_HELP: Specifies the channel configuration in the uplink signal.↳
↳This command has the same effect as the sum of the following commands:

- method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.
↳Dpcch.set
- method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.
↳Dpdch.set
- method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.
↳Hsdpcch.set
- method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.
↳Edpcch.set
```

(continues on next page)

(continued from previous page)

```
- CONFIGure:WCDma:MEAS<i>:MEvaluation:LIMit:RCDerror:EECDp:CARRier<c>:EDPDch
↪<no>
```

Please refer to these commands for additional information (ranges and *RST values) . The parameter array described below is repeated for each channel (eight times) in the following order: DPCCH, DPDCH, HS-DPCCH, E-DPCCH, E-DPDCH 1, ... , E-DPDCH 4. Thus a setting requires 3*8 values and a query returns 5*8 values.

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

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

Cloning the Group

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

Subgroups

7.2.6.2.1.3 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.configure.multiEval.limit.rcdError.eecdpcarrier.edpdch.repcap_edpdChannel_
↪get()
driver.configure.multiEval.limit.rcdError.eecdpcarrier.edpdch.repcap_edpdChannel_
↪set(repcap.EdpdChannel.Nr1)
```

SCPI Commands

```
CONFIGure:WCDma:MEASurement<Instance>:MEvaluation:LIMit:RCDerror:EECDp:CARRier<Carrier>
↪:EDPDch<EdpdChannel>
```

class Edpdch

Edpdch commands group definition. 1 total commands, 0 Sub-groups, 1 group commands Repeated Capability: EdpdChannel, default value after init: EdpdChannel.Nr1

class GetStruct

Response structure. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 5 to 5655
- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256
- Nominal_Cdp: float: float Range: -70 dB to 0 dB, Unit: dB
- Effective_Cdp: float: float Range: -90 dB to 0 dB, Unit: dB

get(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → GetStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>:EDPDch<nr>
value: GetStruct = driver.configure.multiEval.limit.rcdError.eecdP.carrier.
↳edpdch.get(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Specifies the presence of a selected E-DPDCH (1 to 4) in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

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

set(*enable*: bool, *beta_factor*: int, *spreading_factor*: int, *carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>:EDPDch<nr>
driver.configure.multiEval.limit.rcdError.eecdP.carrier.edpdch.set(enable =
↳False, beta_factor = 1, spreading_factor = 1, carrier = repcap.Carrier.
↳Default, edpdChannel = repcap.EdpdChannel.Default)
```

Specifies the presence of a selected E-DPDCH (1 to 4) in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings.

param enable OFF | ON Channel disabled | enabled

param beta_factor numeric Range: 5 to 5655

param spreading_factor numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

Cloning the Group

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

7.2.6.2.1.4 Edpcch

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:MEValuation:LIMit:RCDerror:EECDp:CARRier<Carrier>
↳:EDPCch
```

class Edpcch

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

class GetStruct

Response structure. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 5 to 3585
- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256
- Nominal_Cdp: float: float Range: -70 dB to 0 dB, Unit: dB
- Effective_Cdp: float: float Range: -90 dB to 0 dB, Unit: dB

get(carrier=<Carrier.Default: -1>) → GetStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:MEValuation:LIMit:RCDerror:EECDp:CARRier<carrier>:EDPCch
value: GetStruct = driver.configure.multiEval.limit.rcdError.eecdP.carrier.
↳edpcch.get(carrier = repcap.Carrier.Default)
```

Specifies the presence of an E-DPCCH in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

set(enable: bool, beta_factor: int, spreading_factor: int, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:MEValuation:LIMit:RCDerror:EECDp:CARRier<carrier>:EDPCch
driver.configure.multiEval.limit.rcdError.eecdP.carrier.edpcch.set(enable = False,
↳beta_factor = 1, spreading_factor = 1, carrier = repcap.Carrier.
↳Default)
```

Specifies the presence of an E-DPCCH in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings.

param enable OFF | ON Channel disabled | enabled

param beta_factor numeric Range: 5 to 3585

param spreading_factor numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.6.2.1.5 Hsdpcch

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIMit:RCDerror:EECDp:CARRier<Carrier>
↳:HSDPcch
```

class Hsdpcch

Hsdpcch commands group definition. 2 total commands, 1 Sub-groups, 1 group commands

class GetStruct

Response structure. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 5 to 570
- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256
- Nominal_Cdp: float: float Range: -70 dB to 0 dB, Unit: dB
- Effective_Cdp: float: float Range: -90 dB to 0 dB, Unit: dB

get(carrier=<Carrier.Default: -1>) → GetStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:MEValuation:LIMit:RCDerror:EECDp:CARRier<carrier>:HSDPcch
value: GetStruct = driver.configure.multiEval.limit.rcdError.eecdP.carrier.
↳hsdpcch.get(carrier = repcap.Carrier.Default)
```

Specifies the presence of an HS-DPCCH in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings. For the HS-DPCCH three sets of beta factor and spreading factor can be configured, depending on whether it transports an ACK, NACK or CQI. This command configures/returns the values related to the currently active set. For selection of the active set, see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.Hsdpcch.Config.set.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

set(enable: bool, beta_factor: int, spreading_factor: int, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:MEValuation:LIMit:RCDerror:EECDp:CARRier<carrier>:HSDPcch
driver.configure.multiEval.limit.rcdError.eecdP.carrier.hsdpcch.set(enable =
↳False, beta_factor = 1, spreading_factor = 1, carrier = repcap.Carrier.
↳Default)
```

Specifies the presence of an HS-DPCCH in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings. For the HS-DPCCH three sets of beta factor and spreading factor can be configured, depending on whether it transports an ACK, NACK or CQI. This command configures/returns the values related to the currently active set. For selection of the active set, see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.RcdError.EecdP.Carrier.Hsdpcch.Config.set.

param enable OFF | ON Channel disabled | enabled

param beta_factor numeric Range: 5 to 570

param spreading_factor numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

Cloning the Group

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

Subgroups

7.2.6.2.1.6 Config

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:RCDError:EECDp:CARRier<Carrier>
↳:HSDPcch:CONFig
```

class Config

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

get(carrier=<Carrier.Default: -1>) → RsCmwWcdmaMeas.enums.Type

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>:HSDPcch:CONFig
value: enums.Type = driver.configure.multiEval.limit.rcdError.eecdpcarrier.
↳hsdpcch.config.get(carrier = repcap.Carrier.Default)
```

Selects whether the HS-DPCCH transports an ACK, NACK or CQI and thus which set of beta factor and spreading factor values is used.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return type_py: ACK | NACK | CQI

set(type_py: RsCmwWcdmaMeas.enums.Type, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>:HSDPcch:CONFig
driver.configure.multiEval.limit.rcdError.eecdpcarrier.hsdpcch.config.set(type_
↳py = enums.Type.ACK, carrier = repcap.Carrier.Default)
```

Selects whether the HS-DPCCH transports an ACK, NACK or CQI and thus which set of beta factor and spreading factor values is used.

param type_py ACK | NACK | CQI

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.6.2.1.7 Dpdch

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:RCDerror:EECDp:CARRier<Carrier>
↳:DPDCh
```

class Dpdch

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

class GetStruct

Response structure. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 0 to 15
- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256
- Nominal_Cdp: float: float Range: -60 dB to 0 dB, Unit: dB
- Effective_Cdp: float: float Range: -80 dB to 0 dB, Unit: dB

get(carrier=<Carrier.Default: -1>) → GetStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>:DPDCh
value: GetStruct = driver.configure.multiEval.limit.rcdError.eecdP.carrier.
↳dpdch.get(carrier = repcap.Carrier.Default)
```

Specifies the presence of a DPDCH in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

set(enable: bool, beta_factor: int, spreading_factor: int, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>:DPDCh
driver.configure.multiEval.limit.rcdError.eecdP.carrier.dpdch.set(enable =
↳False, beta_factor = 1, spreading_factor = 1, carrier = repcap.Carrier.
↳Default)
```

Specifies the presence of a DPDCH in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings.

param enable OFF | ON Channel disabled | enabled

param beta_factor numeric Range: 0 to 15

param spreading_factor numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

7.2.6.2.1.8 Dpcch

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:MEvaluation:LIMit:RCDerror:EECDp:CARRier<Carrier>
↳:DPCCh
```

class Dpcch

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

class GetStruct

Response structure. Fields:

- Enable: bool: OFF | ON Channel disabled | enabled
- Beta_Factor: int: numeric Range: 1 to 15
- Spreading_Factor: int: numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256
- Nominal_Cdp: float: float Range: -60 dB to 0 dB, Unit: dB
- Effective_Cdp: float: float Range: -80 dB to 0 dB, Unit: dB

get(carrier=<Carrier.Default: -1>) → GetStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>:DPCCh
value: GetStruct = driver.configure.multiEval.limit.rcdError.eecdP.carrier.
↳dpcch.get(carrier = repcap.Carrier.Default)
```

Specifies the presence of a DPCCH in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

set(enable: bool, beta_factor: int, spreading_factor: int, carrier=<Carrier.Default: -1>) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>:DPCCh
driver.configure.multiEval.limit.rcdError.eecdP.carrier.dpcch.set(enable =
↳False, beta_factor = 1, spreading_factor = 1, carrier = repcap.Carrier.
↳Default)
```

Specifies the presence of a DPCCH in the uplink signal and the beta factor and spreading factor of the channel. A query returns also the nominal CDP and effective CDP resulting from these settings.

param enable OFF | ON Channel disabled | enabled

param beta_factor numeric Range: 1 to 15

param spreading_factor numeric Range: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256

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

7.2.6.2.2 Pcontrol

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIMit:PCONtrol:HSDPcch
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIMit:PCONtrol:EPSTep
```

class Pcontrol

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

class EpStepStruct

Structure for reading output parameters. Fields:

- Expected_0_Db: float: numeric Tolerance value for power step size 0 dB Range: 0 dB to 5 dB, Unit: dB
- Expected_1_Db: float: numeric Tolerance value for power step size 1 dB Range: 0 dB to 5 dB, Unit: dB
- Expected_2_Db: float: numeric Tolerance value for power step size 2 dB Range: 0 dB to 5 dB, Unit: dB
- Expected_3_Db: float: numeric Tolerance value for power step size 3 dB Range: 0 dB to 5 dB, Unit: dB
- Expected_4_To_7_Db: float: numeric Tolerance value for power step size 4 dB to 7 dB Range: 0 dB to 5 dB, Unit: dB

class HsdpcchStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Dtx_To_Nack: float: numeric Range: -10 dB to 10 dB, Unit: dB
- Nack_To_Cqi: float: numeric Range: -10 dB to 10 dB, Unit: dB
- Cqi_To_Dtx: float: numeric Range: -10 dB to 10 dB, Unit: dB
- Test_Case: enums.TestCase: T0DB | T1DB T0DB: measurement below maximum UE power with TPC command = 0 dB T1DB: measurement at maximum UE power with TPC command = 1 dB

get_ep_step() → EpStepStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PCONtrol:EPSTep
value: EpStepStruct = driver.configure.multiEval.limit.pcontrol.get_ep_step()
```

Defines tolerance values (‘Expected Power Step Limits’) depending on the nominal power step size.

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

get_hsdpcch() → HsdpcchStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PCONtrol:HSDPcch
value: HsdpcchStruct = driver.configure.multiEval.limit.pcontrol.get_hsdpcch()
```

Defines nominal power steps for the HS-DPCCH limit set. Measurements at maximum UE power and below maximum UE power are supported. Separate values can be defined for the boundaries DTX > (N) ACK, (N) ACK > CQI and CQI > DTX. Also the limit check can be enabled or disabled. See also ‘Power Control Limits’

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

set_ep_step(value: RsCmwWcdmaMeas.Implementations.Configure_.MultiEval_.Limit_.Pcontrol.Pcontrol.EpStepStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PCONtrol:EPSTep
driver.configure.multiEval.limit.pcontrol.set_ep_step(value = EpStepStruct())
```

Defines tolerance values (‘Expected Power Step Limits’) depending on the nominal power step size.

param value see the help for EpStepStruct structure arguments.

set_hsdpcch(value: RsCmwWcdmaMeas.Implementations.Configure_.MultiEval_.Limit_.Pcontrol.Pcontrol.HsdpcchStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PCONtrol:HSDPcch
driver.configure.multiEval.limit.pcontrol.set_hsdpcch(value = HsdpcchStruct())
```

Defines nominal power steps for the HS-DPCCH limit set. Measurements at maximum UE power and below maximum UE power are supported. Separate values can be defined for the boundaries DTX > (N) ACK, (N) ACK > CQI and CQI > DTX. Also the limit check can be enabled or disabled. See also ‘Power Control Limits’

param value see the help for HsdpcchStruct structure arguments.

7.2.6.2.3 Emask

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIMit:EMASK:ABSolute
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIMit:EMASK:RELative
```

class Emask

Emask commands group definition. 3 total commands, 1 Sub-groups, 2 group commands

class AbsoluteStruct

Structure for reading output parameters. Fields:

- Limit_G_3_M_84: float or bool: numeric | ON | OFF Absolute limit line G referenced to a 3.84 MHz filter Range: -80 dBm to 33 dBm, Unit: dBm Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Limit_H_1_Mhz: float or bool: numeric | ON | OFF Absolute limit line H referenced to a 1 MHz or 100 kHz filter, depending on the line H mode Range: -80 dBm to 33 dBm, Unit: dBm Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Limit_H_30_Khz: float or bool: numeric | ON | OFF Absolute limit line H referenced to a 30 kHz filter Range: -80 dBm to 33 dBm Additional OFF | ON disables/enables the limit check using the previous/default limit values

- Limit_Hmode: enums.LimitHmode: A | B | C Line H mode

class RelativeStruct

Structure for reading output parameters. Fields:

- Point_A: float or bool: numeric | ON | OFF Range: -90 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Point_B: float or bool: numeric | ON | OFF Range: -90 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Point_C: float or bool: numeric | ON | OFF Range: -90 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Point_D: float or bool: numeric | ON | OFF Range: -90 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Point_E: float or bool: numeric | ON | OFF Range: -90 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Point_F: float or bool: numeric | ON | OFF Range: -90 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values

get_absolute() → AbsoluteStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EMASk:ABSolute
value: AbsoluteStruct = driver.configure.multiEval.limit.emask.get_absolute()
```

Defines absolute limits for the spectrum emission curves.

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

get_relative() → RelativeStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EMASk:RELative
value: RelativeStruct = driver.configure.multiEval.limit.emask.get_relative()
```

Defines relative limits for the spectrum emission curves.

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

set_absolute(value: RsCmwWcd-
maMeas.Implementations.Configure_.MultiEval_.Limit_.Emask.Emask.AbsoluteStruct) →
None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EMASk:ABSolute
driver.configure.multiEval.limit.emask.set_absolute(value = AbsoluteStruct())
```

Defines absolute limits for the spectrum emission curves.

param value see the help for AbsoluteStruct structure arguments.

set_relative(value: RsCmwWcd-
maMeas.Implementations.Configure_.MultiEval_.Limit_.Emask.Emask.RelativeStruct) →
None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIMit:EMASk:RELative
driver.configure.multiEval.limit.emask.set_relative(value = RelativeStruct())
```

Defines relative limits for the spectrum emission curves.

param value see the help for RelativeStruct structure arguments.

Cloning the Group

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

Subgroups

7.2.6.2.3.1 Dcarrier

SCPI Commands

```
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:LIMit:EMASk:DCARrier:ABSolute
```

class Dcarrier

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

class AbsoluteStruct

Structure for reading output parameters. Fields:

- Point_Ij: float or bool: numeric | ON | OFF Absolute limit line I-J referenced to a 1 MHz filter. Range: -80 dBm to 33 dBm, Unit: dBm Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Point_Jk: float or bool: numeric | ON | OFF Absolute limit line J-K referenced to a 1 MHz filter. Range: -80 dBm to 33 dBm, Unit: dBm Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Point_Kl: float or bool: numeric | ON | OFF Absolute limit line K-L referenced to a 1 MHz filter. Range: -80 dBm to 33 dBm, Unit: dBm Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Point_Mn: float or bool: numeric | ON | OFF Absolute limit line M-N referenced to a 30 kHz filter. Range: -80 dBm to 33 dBm, Unit: dBm Additional OFF | ON disables/enables the limit check using the previous/default limit values

get_absolute() → AbsoluteStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↪:MEValuation:LIMit:EMASk:DCARrier:ABSolute
value: AbsoluteStruct = driver.configure.multiEval.limit.emask.dcarrier.get_
↪absolute()
```

Defines absolute limits for the spectrum emission curves of DC HSPA connections.

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

```
set_absolute(value: RsCmwWcd-
             maMeas.Implementations.Configure_.MultiEval_.Limit_.Emask_.Dcarrier.Dcarrier.AbsoluteStruct)
             → None
```

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↪:MEValuation:LIMit:EMASk:DCARrier:ABSolute
driver.configure.multiEval.limit.emask.dcarrier.set_absolute(value =
↪AbsoluteStruct())
```

Defines absolute limits for the spectrum emission curves of DC HSPA connections.

param value see the help for AbsoluteStruct structure arguments.

7.2.6.2.4 Aclr

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIMit:ACLR:RELative
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIMit:ACLR:ABSolute
```

class Aclr

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

class RelativeStruct

Structure for reading output parameters. Fields:

- Channel_First: float or bool: numeric | ON | OFF For single uplink carrier: ± 5 MHz from the center frequency For dual uplink carrier: ± 7.5 MHz from the center frequency of both carriers Range: -80 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values
- Channel_Second: float or bool: numeric | ON | OFF For single uplink carrier: ± 10 MHz from the center frequency For dual uplink carrier: ± 12.5 MHz from the center frequency of both carriers Range: -80 dB to 0 dB, Unit: dB Additional OFF | ON disables/enables the limit check using the previous/default limit values

get_absolute() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:ACLR:ABSolute
value: float or bool = driver.configure.multiEval.limit.aclr.get_absolute()
```

Defines an absolute upper limit for the ACLR. If the absolute upper limit is exceeded, relative limits are evaluated (method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Aclr.relative) .

return limit_3_m_84: numeric | ON | OFF Range: -80 dBm to 33 dBm, Unit: dBm
Additional OFF | ON disables/enables the limit check using the previous/default limit values

get_relative() → RelativeStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:ACLR:RELative
value: RelativeStruct = driver.configure.multiEval.limit.aclr.get_relative()
```


Defines upper limits for the ACLR in channels one and two relative to the carrier power. Relative limits are only evaluated when the absolute limit is exceeded (method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Aclr.absolute) .

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

set_absolute(*limit_3_m_84: float*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:ACLR:ABSolute
driver.configure.multiEval.limit.aclr.set_absolute(limit_3_m_84 = 1.0)
```

Defines an absolute upper limit for the ACLR. If the absolute upper limit is exceeded, relative limits are evaluated (method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Aclr.relative) .

param limit_3_m_84 numeric | ON | OFF Range: -80 dBm to 33 dBm, Unit: dBm
Additional OFF | ON disables/enables the limit check using the previous/default limit values

set_relative(*value: RsCmwWcdmaMeas.Implementations.Configure_.MultiEval_.Limit_.Aclr.Aclr.RelativeStruct*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:ACLR:RELative
driver.configure.multiEval.limit.aclr.set_relative(value = RelativeStruct())
```

Defines upper limits for the ACLR in channels one and two relative to the carrier power. Relative limits are only evaluated when the absolute limit is exceeded (method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Aclr.absolute) .

param value see the help for RelativeStruct structure arguments.

7.2.6.3 Rotation

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:ROTation:MODulation
```

class Rotation

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

get_modulation() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:ROTation:MODulation
value: int = driver.configure.multiEval.rotation.get_modulation()
```

Defines the initial phase reference (=0) for I/Q constellation diagrams of QPSK signals.

return rotation: numeric The entered value is rounded to 0 deg or 45 deg. 0 deg: constellation points on I- and Q-axes 45 deg: constellation points on angle bisectors between I- and Q-axes Range: 0 deg to 45 deg, Unit: deg

set_modulation(*rotation: int*) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:ROtation:MODulation
driver.configure.multiEval.rotation.set_modulation(rotation = 1)
```

Defines the initial phase reference (=0) for I/Q constellation diagrams of QPSK signals.

param rotation numeric The entered value is rounded to 0 deg or 45 deg. 0 deg: constellation points on I- and Q-axes 45 deg: constellation points on angle bisectors between I- and Q-axes Range: 0 deg to 45 deg, Unit: deg

7.2.6.4 DsFactor

SCPI Commands

```
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:DSFactor:MODulation
```

class DsFactor

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

get_modulation() → RsCmwWcdmaMeas.enums.SpreadingFactorA

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:DSFactor:MODulation
value: enums.SpreadingFactorA = driver.configure.multiEval.dsFactor.get_
↪modulation()
```

Selects the spreading factor for the displayed code domain monitor results.

return spreading_factor: SF4 | SF8 | SF16 | SF32 | SF64 | SF128 | SF256 Spreading factor 4 to 256

set_modulation(spreading_factor: RsCmwWcdmaMeas.enums.SpreadingFactorA) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:DSFactor:MODulation
driver.configure.multiEval.dsFactor.set_modulation(spreading_factor = enums.
↪SpreadingFactorA.SF128)
```

Selects the spreading factor for the displayed code domain monitor results.

param spreading_factor SF4 | SF8 | SF16 | SF32 | SF64 | SF128 | SF256 Spreading factor 4 to 256

7.2.6.5 Sscalar

SCPI Commands

```
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:SScalar:MODulation
```

class Sscalar

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

get_modulation() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SSCalAr:MODulation
value: float = driver.configure.multiEval.sscalar.get_modulation()
```

Selects a particular slot or half-slot within the measurement length where the R&S CMW evaluates the statistical measurement results for multislot measurements. The slot number must be smaller than the number of measured slots (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

return slot_number: numeric Range: 0 to 119.5

set_modulation(slot_number: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SSCalAr:MODulation
driver.configure.multiEval.sscalar.set_modulation(slot_number = 1.0)
```

Selects a particular slot or half-slot within the measurement length where the R&S CMW evaluates the statistical measurement results for multislot measurements. The slot number must be smaller than the number of measured slots (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

param slot_number numeric Range: 0 to 119.5

7.2.6.6 CdThreshold

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:CDTHreshold:MODulation
```

class CdThreshold

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

get_modulation() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:CDTHreshold:MODulation
value: float = driver.configure.multiEval.cdThreshold.get_modulation()
```

Defines the minimum relative signal strength of the (E-) DPDCH in the WCDMA signal (if present) to be detected and evaluated.

return threshold: numeric Range: -25 dB to 10 dB, Unit: dB

set_modulation(threshold: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:CDTHreshold:MODulation
driver.configure.multiEval.cdThreshold.set_modulation(threshold = 1.0)
```

Defines the minimum relative signal strength of the (E-) DPDCH in the WCDMA signal (if present) to be detected and evaluated.

param threshold numeric Range: -25 dB to 10 dB, Unit: dB

7.2.6.7 Dmode

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:MEValuation:DMODE:MODulation
```

class Dmode

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

get_modulation() → RsCmwWcdmaMeas.enums.DetectionMode

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:DMODE:MODulation
value: enums.DetectionMode = driver.configure.multiEval.dmode.get_modulation()
```

Selects the detection mode for uplink WCDMA signals.

return detection_mode: A3G A3G: '3GPP Signal Auto'

set_modulation(detection_mode: RsCmwWcdmaMeas.enums.DetectionMode) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:DMODE:MODulation
driver.configure.multiEval.dmode.set_modulation(detection_mode = enums.
↳DetectionMode.A3G)
```

Selects the detection mode for uplink WCDMA signals.

param detection_mode A3G A3G: '3GPP Signal Auto'

7.2.6.8 Amode

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:MEValuation:AMODE:MODulation
```

class Amode

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

get_modulation() → RsCmwWcdmaMeas.enums.AnalysisMode

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:AMODE:MODulation
value: enums.AnalysisMode = driver.configure.multiEval.amode.get_modulation()
```

Defines whether a possible origin offset is included in the measurement results (WOOFset) or subtracted out (NOOFset) .

return analysis_mode: WOOFset | NOOFset WOOFset: With origin offset NOOFset: No origin offset

set_modulation(analysis_mode: RsCmwWcdmaMeas.enums.AnalysisMode) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:AMODE:MODulation
driver.configure.multiEval.amode.set_modulation(analysis_mode = enums.
↳AnalysisMode.NOOFset)
```

Defines whether a possible origin offset is included in the measurement results (WOOFset) or subtracted out (NOOFset) .

param analysis_mode WOOFset | NOOFset WOOFset: With origin offset NOOFset: No origin offset

7.2.6.9 Mperiod

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:MPERiod:MODulation
```

class Mperiod

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

get_modulation() → RsCmwWcdmaMeas.enums.MeasPeriod

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:MPERiod:MODulation
value: enums.MeasPeriod = driver.configure.multiEval.mperiod.get_modulation()
```

Selects the width of the basic measurement period within each measured slot. To define the number of measured slots, see method RsCmwWcdmaMeas.Configure.MultiEval.msCount.

return meas_period: FULLslot | HALFslot FULLslot: Full-slot measurement HALFslot: Half-slot measurement

set_modulation(meas_period: RsCmwWcdmaMeas.enums.MeasPeriod) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:MPERiod:MODulation
driver.configure.multiEval.mperiod.set_modulation(meas_period = enums.
    ↪ MeasPeriod.FULLslot)
```

Selects the width of the basic measurement period within each measured slot. To define the number of measured slots, see method RsCmwWcdmaMeas.Configure.MultiEval.msCount.

param meas_period FULLslot | HALFslot FULLslot: Full-slot measurement HALFslot: Half-slot measurement

7.2.6.10 Result

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:TXM
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:RCDerror
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:IQ
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:BER
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:PSTeps
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:PHD
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:FERRor
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:UEPower
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:ALL
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:CDERRor
```

(continues on next page)

(continued from previous page)

```

CONFIGure:WCDma:MEASurement<Instance>:MEvaluation:RESult:CDPower
CONFIGure:WCDma:MEASurement<Instance>:MEvaluation:RESult:CDPMonitor
CONFIGure:WCDma:MEASurement<Instance>:MEvaluation:RESult:EMASk
CONFIGure:WCDma:MEASurement<Instance>:MEvaluation:RESult:ACLR
CONFIGure:WCDma:MEASurement<Instance>:MEvaluation:RESult:PERRor
CONFIGure:WCDma:MEASurement<Instance>:MEvaluation:RESult:EVMagnitude
CONFIGure:WCDma:MEASurement<Instance>:MEvaluation:RESult:MERRor

```

class Result

Result commands group definition. 20 total commands, 1 Sub-groups, 17 group commands

class AllStruct

Structure for reading output parameters. Fields:

- Enable_Evm: bool: OFF | ON Error vector magnitude OFF: Do not evaluate results, hide the view
ON: Evaluate results and show the view
- Enable_Mag_Error: bool: OFF | ON Magnitude error
- Enable_Phase_Err: bool: OFF | ON Phase error
- Enable_Aclr: bool: OFF | ON Adjacent channel leakage power ratio
- Enable_Emask: bool: OFF | ON Spectrum emission mask
- Enable_Cd_Monitor: bool: OFF | ON Code domain monitor
- Enable_Cdp: bool: OFF | ON Code domain power
- Enable_Cde: bool: OFF | ON Code domain error
- Enable_Evm_Chip: bool: OFF | ON EVM vs. chip
- Enable_Merr_Chip: bool: OFF | ON Magnitude error vs. chip
- Enable_Ph_Err_Chip: bool: OFF | ON Phase error vs. chip
- Enable_Ue_Power: bool: OFF | ON UE power
- Enable_Freq_Error: bool: OFF | ON Frequency error
- Enable_Phase_Disc: bool: OFF | ON Phase discontinuity
- Enable_Pow_Steps: bool: OFF | ON Power steps
- Enable_Ber: bool: OFF | ON Bit error rate
- Enable_Iq: bool: OFF | ON I/Q constellation diagram
- Enable_Rcde: bool: OFF | ON Relative CDE
- Enable_Txm: bool: No parameter help available

get_aclr() → bool

```

# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEvaluation:RESult:ACLR
value: bool = driver.configure.multiEval.result.get_aclr()

```

Enables or disables the evaluation of results and shows or hides the adjacent channel leakage power ratio view in the multi-evaluation measurement.

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

get_all() → AllStruct

```
# SCPI: CONFIGure:WCDMa: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. This command combines all other CONFIGure:WCDMa:MEAS<i>:MEValuation:RESult... commands.

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

get_ber() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:BER
value: bool = driver.configure.multiEval.result.get_ber()
```

Enables or disables the evaluation of results and shows or hides the bit error rate view in the multi-evaluation measurement.

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

get_cd_error() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CDError
value: bool = driver.configure.multiEval.result.get_cd_error()
```

Enables or disables the evaluation of results and shows or hides the code domain error view in the multi-evaluation measurement.

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

get_cd_power() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CDPower
value: bool = driver.configure.multiEval.result.get_cd_power()
```

Enables or disables the evaluation of results and shows or hides the code domain power view in the multi-evaluation measurement.

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

get_cdp_monitor() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CDPMonitor
value: bool = driver.configure.multiEval.result.get_cdp_monitor()
```

Enables or disables the evaluation of results and shows or hides the code domain monitor view in the multi-evaluation measurement.

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

get_emask() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:EMASk
value: bool = driver.configure.multiEval.result.get_emask()
```

Enables or disables the evaluation of results and shows or hides the spectrum emission mask view in the multi-evaluation measurement.

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

get_ev_magnitude() → bool

```
# SCPI: CONFIGure:WCDMa: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 error vector magnitude view in the multi-evaluation measurement.

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

get_freq_error() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:FERRor
value: bool = driver.configure.multiEval.result.get_freq_error()
```

Enables or disables the evaluation of results and shows or hides the frequency error view in the multi-evaluation measurement.

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

get_iq() → bool

```
# SCPI: CONFIGure:WCDMa: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 I/Q constellation diagram view in the multi-evaluation measurement.

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

get_merror() → bool

```
# SCPI: CONFIGure:WCDMa: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 magnitude error view in the multi-evaluation measurement.

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

get_perror() → bool

```
# SCPI: CONFIGure:WCDMa: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 phase error view in the multi-evaluation measurement.

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

get_phd() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:PHD
value: bool = driver.configure.multiEval.result.get_phd()
```

Enables or disables the evaluation of results and shows or hides the phase discontinuity view in the multi-evaluation measurement.

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

get_psteps() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:PSTeps
value: bool = driver.configure.multiEval.result.get_psteps()
```

Enables or disables the evaluation of results and shows or hides the power steps view in the multi-evaluation measurement.

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

get_rcd_error() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:RCDerror
value: bool = driver.configure.multiEval.result.get_rcd_error()
```

Enables or disables the evaluation of results and shows or hides the relative CDE view in the multi-evaluation measurement.

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

get_txm() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:TXM
value: bool = driver.configure.multiEval.result.get_txm()
```

No command help available

return enable_txm: No help available

get_ue_power() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:UEPower
value: bool = driver.configure.multiEval.result.get_ue_power()
```

Enables or disables the evaluation of results and shows or hides the UE power view in the multi-evaluation measurement.

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

set_aclr(enable_aclr: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:ACLR
driver.configure.multiEval.result.set_aclr(enable_aclr = False)
```

Enables or disables the evaluation of results and shows or hides the adjacent channel leakage power ratio view in the multi-evaluation measurement.

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

set_all(value: RsCmwWcdmaMeas.Implementations.Configure_.MultiEval_.Result.Result.AllStruct) → None

```
# SCPI: CONFIGure:WCDMa: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. This command combines all other CONFIGure:WCDMa:MEAS<i>:MEValuation:RESult... commands.

param value see the help for AllStruct structure arguments.

set_ber(enable_ber: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:BER
driver.configure.multiEval.result.set_ber(enable_ber = False)
```

Enables or disables the evaluation of results and shows or hides the bit error rate view in the multi-evaluation measurement.

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

set_cd_error(enable_cde: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CDError
driver.configure.multiEval.result.set_cd_error(enable_cde = False)
```

Enables or disables the evaluation of results and shows or hides the code domain error view in the multi-evaluation measurement.

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

set_cd_power(*enable_cdp: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CDPower
driver.configure.multiEval.result.set_cd_power(enable_cdp = False)
```

Enables or disables the evaluation of results and shows or hides the code domain power view in the multi-evaluation measurement.

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

set_cdp_monitor(*enable_cd_monitor: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CDPMonitor
driver.configure.multiEval.result.set_cdp_monitor(enable_cd_monitor = False)
```

Enables or disables the evaluation of results and shows or hides the code domain monitor view in the multi-evaluation measurement.

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

set_emask(*enable_emask: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:EMASK
driver.configure.multiEval.result.set_emask(enable_emask = False)
```

Enables or disables the evaluation of results and shows or hides the spectrum emission mask view in the multi-evaluation measurement.

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

set_ev_magnitude(*enable_evm: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:EVMagnitude
driver.configure.multiEval.result.set_ev_magnitude(enable_evm = False)
```

Enables or disables the evaluation of results and shows or hides the error vector magnitude view in the multi-evaluation measurement.

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

set_freq_error(*enable_freq_error: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:FERRor
driver.configure.multiEval.result.set_freq_error(enable_freq_error = False)
```

Enables or disables the evaluation of results and shows or hides the frequency error view in the multi-evaluation measurement.

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

set_iq(*enable_iq: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:IQ
driver.configure.multiEval.result.set_iq(enable_iq = False)
```

Enables or disables the evaluation of results and shows or hides the I/Q constellation diagram view in the multi-evaluation measurement.

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

set_merror(*enable_mag_error: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:MERRor
driver.configure.multiEval.result.set_merror(enable_mag_error = False)
```

Enables or disables the evaluation of results and shows or hides the magnitude error view in the multi-evaluation measurement.

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

set_perror(*enable_phase_err: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:PERRor
driver.configure.multiEval.result.set_perror(enable_phase_err = False)
```

Enables or disables the evaluation of results and shows or hides the phase error view in the multi-evaluation measurement.

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

set_phd(*enable_phase_disc: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:PHD
driver.configure.multiEval.result.set_phd(enable_phase_disc = False)
```

Enables or disables the evaluation of results and shows or hides the phase discontinuity view in the multi-evaluation measurement.

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

set_psteps(*enable_pow_steps: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:PSTeps
driver.configure.multiEval.result.set_psteps(enable_pow_steps = False)
```

Enables or disables the evaluation of results and shows or hides the power steps view in the multi-evaluation measurement.

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

set_rcd_error(enable_rcde: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:RCDError
driver.configure.multiEval.result.set_rcd_error(enable_rcde = False)
```

Enables or disables the evaluation of results and shows or hides the relative CDE view in the multi-evaluation measurement.

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

set_txm(enable_txm: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:TXM
driver.configure.multiEval.result.set_txm(enable_txm = False)
```

No command help available

param enable_txm No help available

set_ue_power(enable_ue_power: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:UEPower
driver.configure.multiEval.result.set_ue_power(enable_ue_power = False)
```

Enables or disables the evaluation of results and shows or hides the UE power view in the multi-evaluation measurement.

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

Cloning the Group

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

Subgroups

7.2.6.10.1 Chip

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:CHIP:PERRor
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:CHIP:MERRor
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:RESult:CHIP:EVM
```

class Chip

Chip commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

get_evm() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CHIP:EVM
value: bool = driver.configure.multiEval.result.chip.get_evm()
```

Enables or disables the evaluation of results and shows or hides the EVM vs. chip view in the multi-evaluation measurement.

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

get_merror() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CHIP:MERRor
value: bool = driver.configure.multiEval.result.chip.get_merror()
```

Enables or disables the evaluation of results and shows or hides the magnitude error vs. chip view in the multi-evaluation measurement.

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

get_perror() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CHIP:PERRor
value: bool = driver.configure.multiEval.result.chip.get_perror()
```

Enables or disables the evaluation of results and shows or hides the phase error vs. chip view in the multi-evaluation measurement.

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

set_evm(enable_evm_chip: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CHIP:EVM
driver.configure.multiEval.result.chip.set_evm(enable_evm_chip = False)
```

Enables or disables the evaluation of results and shows or hides the EVM vs. chip view in the multi-evaluation measurement.

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

set_merror(enable_merr_chip: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult:CHIP:MERRor
driver.configure.multiEval.result.chip.set_merror(enable_merr_chip = False)
```

Enables or disables the evaluation of results and shows or hides the magnitude error vs. chip view in the multi-evaluation measurement.

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

set_perror(enable_ph_err_chip: bool) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:RESult:CHIP:PERRor
driver.configure.multiEval.result.chip.set_perror(enable_ph_err_chip = False)
```

Enables or disables the evaluation of results and shows or hides the phase error vs. chip view in the multi-evaluation measurement.

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

7.2.6.11 ListPy

SCPI Commands

```
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:LIST:EOffset
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:LIST:COUNt
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:LIST:OSINdex
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:LIST
```

class ListPy

ListPy commands group definition. 12 total commands, 2 Sub-groups, 4 group commands

get_count() → int

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:COUNt
value: int = driver.configure.multiEval.listPy.get_count()
```

Defines the number of segments in the entire measurement interval, including active and inactive segments.

return segments: numeric Range: 1 to 1000

get_eoffset() → int

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:EOffset
value: int = driver.configure.multiEval.listPy.get_eoffset()
```

Defines the evaluation offset. The specified number of slots at the beginning of each segment is excluded from the evaluation. Set the trigger delay to 0 when using an evaluation offset (see method RsCmwWcdmaMeas.Trigger.MultiEval.delay).

return offset: numeric Range: 0 slots to 1024 slots

get_os_index() → int

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:OSINdex
value: int = driver.configure.multiEval.listPy.get_os_index()
```

No command help available

return index: No help available

get_value() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST
value: bool = driver.configure.multiEval.listPy.get_value()
```

Enables or disables the list mode.

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

set_count(segments: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:COUNT
driver.configure.multiEval.listPy.set_count(segments = 1)
```

Defines the number of segments in the entire measurement interval, including active and inactive segments.

param segments numeric Range: 1 to 1000

set_eoffset(offset: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:EOffset
driver.configure.multiEval.listPy.set_eoffset(offset = 1)
```

Defines the evaluation offset. The specified number of slots at the beginning of each segment is excluded from the evaluation. Set the trigger delay to 0 when using an evaluation offset (see method RsCmwWcdmaMeas.Trigger.MultiEval.delay).

param offset numeric Range: 0 slots to 1024 slots

set_os_index(index: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:OSIndex
driver.configure.multiEval.listPy.set_os_index(index = 1)
```

No command help available

param index No help available

set_value(enable: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST
driver.configure.multiEval.listPy.set_value(enable = False)
```

Enables or disables the list mode.

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

Cloning the Group

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

Subgroups

7.2.6.11.1 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. 7 total commands, 7 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

7.2.6.11.1.1 Setup

SCPI Commands

```
CONFigure:WCDma: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 timeslots in the segment. The sum of the length of all active segments must not exceed 6000. Ignoring this limit results in NCAPs for the remaining slots. The statistical length for result calculation covers at most the first 1000 slots of a segment. The sum of the length of all segments (active plus inactive) must not exceed 12000. 'Inactive' means that no measurement at all is enabled for the segment. Range: 1 to 12000, Unit: slot
- 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. Unit: dBm
- Frequency: float: numeric Range: 100 MHz to 6 GHz, Unit: Hz

- **Retrigger:** `enums.Retrigger`: Optional setting parameter. OFF | ON | IFPower | IFPSync
Specifies whether a trigger event is required for the segment or not. The setting is ignored for the first segment of a measurement and for trigger mode ONCE (see [CMDLINK: TRIGGER:WCDMA:MEAS:MEVALUATION:LIST:MODE CMDLINK]). OFF: measure the segment without retrigger ON: trigger event required, trigger source configured via [CMDLINK: TRIGGER:WCDMA:MEAS:MEVALUATION:SOURCE CMDLINK] IFPower: trigger event required, 'IF Power' trigger IFPSync: trigger event required, 'IF Power (Sync)' trigger

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

```
# SCPI: CONFIGure:WCDMA:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:SETup
value: SetupStruct = driver.configure.multiEval.listPy.segment.setup.
↪get(segment = repcap.Segment.Default)
```

Defines the length and analyzer settings of a selected segment. 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:

RsCmwWcdmaMeas.Implementations.Configure_.MultiEval_.ListPy_.Segment_.Setup.Setup.SetupStruct,
segment=<Segment.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMA: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 and analyzer settings of a selected segment. 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')

7.2.6.11.1.2 Modulation

SCPI Commands

```
CONFIGure:WCDMA: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. Fields:

- **Mod_Statistics:** int: integer The statistical length is limited by the length of the segment (see [CMDLINK: CONFIGure:WCDMA:MEAS:MEVALUATION:LIST:SEGmentno:SETup CMDLINK]). Range: 1 to 1000
- **Enable_Ue_Power:** bool: OFF | ON OFF: Disable measurement ON: Enable measurement of UE power

- Enable_Evm: bool: OFF | ON Disable or 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_Freq_Error: bool: OFF | ON Disable or enable measurement of frequency error
- Enable_Iq: bool: OFF | ON Disable or enable measurement of I/Q origin offset and imbalance

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

```
# SCPI: CONFIGure:WCDma: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 the AVERage, MAXimum, and SDEViation calculation and enables the calculation of the different modulation results in segment no. <no>; see 'Multi-Evaluation List Mode'. The statistical length for CDP, CDE and modulation results is identical (see also method RsCmwWcdmaMeas.Configure.MultiEval.ListPy.Segment.CdPower.set) .

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: RsCmwWcdmaMeas.Implementations.Configure_.MultiEval_.ListPy_.Segment_.Modulation.Modulation.ModulationStruct, segment=<Segment.Default: -1>) → None

```
# SCPI: CONFIGure:WCDma: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 the AVERage, MAXimum, and SDEViation calculation and enables the calculation of the different modulation results in segment no. <no>; see 'Multi-Evaluation List Mode'. The statistical length for CDP, CDE and modulation results is identical (see also method RsCmwWcdmaMeas.Configure.MultiEval.ListPy.Segment.CdPower.set) .

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')

7.2.6.11.1.3 Spectrum

SCPI Commands

```
CONFIGure:WCDma: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:

- Spec_Statistics: int: integer The statistical length is limited by the length of the segment (see [CMDLINK: CONFigure:WCDMa:MEASi:MEValuation:LIST:SEGMENTno:SETup CMDLINK]) . Range: 1 to 1000
- Enable_Aclr: bool: OFF | ON OFF: Disable measurement ON: Enable measurement of ACLR
- Enable_Emask: bool: OFF | ON Disable or enable measurement of spectrum emission mask
- Enable_Obw: bool: OFF | ON Disable or enable measurement of occupied bandwidth

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

```
# SCPI: CONFigure:WCDMa: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 the AVERage and MAXimum calculation and enables the calculation of the different spectrum results in segment no. <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: RsCmwWcdmaMeas.Implementations.Configure_.MultiEval_.ListPy_.Segment_.Spectrum.Spectrum.SpectrumStruct, segment=<Segment.Default: -1>) → None

```
# SCPI: CONFigure:WCDMa: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 the AVERage and MAXimum calculation and enables the calculation of the different spectrum results in segment no. <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')

7.2.6.11.1.4 CdPower

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGMENT<Segment>:CDPower
```

class CdPower

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

class CdPowerStruct

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

- Mod_Statistics: int: integer The statistical length is limited by the length of the segment (see [CMDLINK: CONFigure:WCDMa:MEASi:MEValuation:LIST:SEGMENTno:SETup CMDLINK]) . Range: 1 to 1000

- Enable_Cdp: bool: OFF | ON OFF: Disable measurement ON: Enable measurement of code domain power
- Enable_Cde: bool: OFF | ON Disable or enable measurement of code domain error
- Enable_Pcde: bool: Optional setting parameter. OFF | ON Disable or enable measurement of peak code domain error

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

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↪:CDPower
value: CdPowerStruct = driver.configure.multiEval.listPy.segment.cdPower.
↪get(segment = repcap.Segment.Default)
```

Defines the statistical length for the AVERage, MINimum, MAXimum and SDEViation calculation and enables the calculation of the different code domain results in segment no. <no>; see 'Multi-Evaluation List Mode'. The statistical length for CDP, CDE, PCDE and modulation results is identical (see also method RsCmwWcdmaMeas.Configure.MultiEval.ListPy.Segment.Modulation.set).

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

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

set(structure: RsCmwWcdmaMeas.Implementations.Configure_.MultiEval_.ListPy_.Segment_.CdPower.CdPower.CdPowerStruct, segment=<Segment.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↪:CDPower
driver.configure.multiEval.listPy.segment.cdPower.set(value = [PROPERTY_STRUCT_
↪NAME](), segment = repcap.Segment.Default)
```

Defines the statistical length for the AVERage, MINimum, MAXimum and SDEViation calculation and enables the calculation of the different code domain results in segment no. <no>; see 'Multi-Evaluation List Mode'. The statistical length for CDP, CDE, PCDE and modulation results is identical (see also method RsCmwWcdmaMeas.Configure.MultiEval.ListPy.Segment.Modulation.set).

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

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

7.2.6.11.1.5 UePower

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:UEPower
```

class UePower

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

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

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:UEPower
value: bool = driver.configure.multiEval.listPy.segment.uePower.get(segment = ↵
↳repcap.Segment.Default)
```

Enables the calculation of the current UE power vs. slot results in segment no. <no>; see ‘Multi-Evaluation List Mode’.

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

return enable_ue_power: OFF | ON OFF: Disable measurement ON: Enable measurement of UE power

set(enable_ue_power: bool, segment=<Segment.Default: -1>) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:UEPower
driver.configure.multiEval.listPy.segment.uePower.set(enable_ue_power = False, ↵
↳segment = repcap.Segment.Default)
```

Enables the calculation of the current UE power vs. slot results in segment no. <no>; see ‘Multi-Evaluation List Mode’.

param enable_ue_power OFF | ON OFF: Disable measurement ON: Enable measurement of UE power

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

7.2.6.11.1.6 Phd

SCPI Commands

```
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:PHD
```

class Phd

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

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

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:PHD
value: bool = driver.configure.multiEval.listPy.segment.phd.get(segment = ↵
↳repcap.Segment.Default)
```

Enables the calculation of the phase discontinuity vs. slot results in segment no. <no>; see ‘Multi-Evaluation List Mode’.

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

return enable_phd: OFF | ON OFF: Disable measurement ON: Enable measurement of phase discontinuity

set(enable_phd: bool, segment=<Segment.Default: -1>) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:PHD
driver.configure.multiEval.listPy.segment.phd.set(enable_phd = False, segment =
↳repcap.Segment.Default)
```

Enables the calculation of the phase discontinuity vs. slot results in segment no. <no>; see ‘Multi-Evaluation List Mode’.

param enable_phd OFF | ON OFF: Disable measurement ON: Enable measurement of phase discontinuity

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

7.2.6.11.1.7 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

7.2.6.11.1.8 Connector

SCPI Commands

```
CONFIGure:WCDma: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>) → RsCmwWcdmaMeas.enums.CmwsConnector

```
# SCPI: CONFIGure:WCDma: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 WCDMA list mode measurements with the R&S CMWS. This setting is only relevant for connector mode LIST, see method RsCmwWcdmaMeas.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: RsCmwWcdmaMeas.enums.CmwsConnector, segment=<Segment.Default: -1>) → None

```
# SCPI: CONFIGure:WCDMa: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 WCDMA list mode measurements with the R&S CMWS. This setting is only relevant for connector mode LIST, see method RsCmwWcdmaMeas.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’)

7.2.6.11.2 SingleCmw

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:CMWS:CMODE
```

class SingleCmw

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

get_cmode() → RsCmwWcdmaMeas.enums.ParameterSetMode

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:CMWS:CMODE
value: enums.ParameterSetMode = driver.configure.multiEval.listPy.singleCmw.get_
↪cmode()
```

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

return 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:WCDMa:MEASi:SCENario:SALone. LIST: The input connector is configured individually for each segment. See method RsCmwWcdmaMeas.Configure.MultiEval.ListPy.Segment.SingleCmw.Connector.set.

set_cmode(connector_mode: RsCmwWcdmaMeas.enums.ParameterSetMode) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:CMWS:CMODE
driver.configure.multiEval.listPy.singleCmw.set_cmode(connector_mode = enums.
↪ParameterSetMode.GLOBAL)
```

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

param 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:WCDMa:MEASi:SCENario:SALone. LIST: The input connector is configured individually for each segment. See method RsCmwWcdmaMeas.Configure.MultiEval.ListPy.Segment.SingleCmw.Connector.set.

7.2.7 Tpc

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:TPC:CSElection
CONFigure:WCDMa:MEASurement<Instance>:TPC:SETup
CONFigure:WCDMa:MEASurement<Instance>:TPC:MODE
CONFigure:WCDMa:MEASurement<Instance>:TPC:MOEXception
CONFigure:WCDMa:MEASurement<Instance>:TPC:TOUT
```

class Tpc

Tpc commands group definition. 32 total commands, 7 Sub-groups, 5 group commands

get_cselection() → RsCmwWcdmaMeas.enums.Carrier

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:CSElection
value: enums.Carrier = driver.configure.tpc.get_cselection()
```

Selects the uplink carrier to be measured. This parameter is relevant only for the dual uplink carrier configuration.

return carrier: C1 | C2 C1: primary uplink carrier C2: secondary uplink carrier

get_mo_exception() → bool

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MOEXception
value: bool = driver.configure.tpc.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 OFF: Faulty results are rejected. ON: Results are never rejected.

get_mode() → RsCmwWcdmaMeas.enums.MeasMode

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MODE
value: enums.MeasMode = driver.configure.tpc.get_mode()
```

Queries the active measurement mode resulting from the currently selected TPC setup.

return meas_mode: MONitor | ILPControl | MPEDch | CTFC | ULCM | DHIB MONitor: 'Monitor' ILPControl: 'Inner Loop Power Control' MPEDch: 'Max. Power E-DCH' CTFC: 'Change of TFC' ULCM: 'UL Compressed Mode' DHIB: 'DC HSPA In-Band Emission'

get_setup() → RsCmwWcdmaMeas.enums.SetType

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:SETup
value: enums.SetType = driver.configure.tpc.get_setup()
```

Selects the TPC setup (expected) to be executed during the measurement. For the combined signal path scenario, use `CONFfigure:WCDMa:SIGN<i>:UL:TPC:SET`.

return `set_type`: CLOop | ALternating | ALL1 | ALL0 | SALT | SAL1 | SAL0 | CONTinuous | TSE | TSF | PHUP | PHDown | TSABc | TSEF | TSGH | MPEDch | ULCM | CTFC | DHIB
 CLOop: 'Closed Loop' ALternating: 'Alternating' ALL1: 'All 1' ALL0: 'All 0' SALT: 'Single Pattern + Alternating' SAL1: 'Single Pattern + All 1' SAL0: 'Single Pattern + All 0' CONTinuous: 'Continuous Pattern' TSE: 'TPC Test Step E' TSF: 'TPC Test Step F' PHUP: 'Phase Discontinuity Up' PHDown: 'Phase Discontinuity Down' TSABc: 'TPC Test Step ABC' TSEF: 'TPC Test Step EF' TSGH: 'TPC Test Step GH' MPEDch: 'Max. Power E-DCH' ULCM: 'TPC Test Step UL CM' CTFC: 'Change of TFC' DHIB: 'DC HSPA In-Band Emission'

get_timeout() → float

```
# SCPI: CONFfigure:WCDMa:MEASurement<instance>:TPC:TOUT
value: float = driver.configure.tpc.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_cselection(carrier: RsCmwWcdmaMeas.enums.Carrier) → None

```
# SCPI: CONFfigure:WCDMa:MEASurement<instance>:TPC:CSElection
driver.configure.tpc.set_cselection(carrier = enums.Carrier.C1)
```

Selects the uplink carrier to be measured. This parameter is relevant only for the dual uplink carrier configuration.

param carrier C1 | C2 C1: primary uplink carrier C2: secondary uplink carrier

set_mo_exception(meas_on_exception: bool) → None

```
# SCPI: CONFfigure:WCDMa:MEASurement<instance>:TPC:MOEXception
driver.configure.tpc.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 OFF: Faulty results are rejected. ON: Results are never rejected.

set_setup(set_type: RsCmwWcdmaMeas.enums.SetType) → None

```
# SCPI: CONFfigure:WCDMa:MEASurement<instance>:TPC:SETup
driver.configure.tpc.set_setup(set_type = enums.SetType.ALL0)
```

Selects the TPC setup (expected) to be executed during the measurement. For the combined signal path scenario, use `CONFIGure:WCDMa:SIGN<i>:UL:TPC:SET`.

param set_type CLOop | ALternating | ALL1 | ALL0 | SALT | SAL1 | SAL0 | CONTinuous | TSE | TSF | PHUP | PHDown | TSABc | TSEF | TSGH | MPEDch | ULCM | CTFC | DHIB
 CLOop: 'Closed Loop' ALternating: 'Alternating' ALL1: 'All 1' ALL0: 'All 0' SALT: 'Single Pattern + Alternating' SAL1: 'Single Pattern + All 1' SAL0: 'Single Pattern + All 0' CONTinuous: 'Continuous Pattern' TSE: 'TPC Test Step E' TSF: 'TPC Test Step F' PHUP: 'Phase Discontinuity Up' PHDown: 'Phase Discontinuity Down' TSABc: 'TPC Test Step ABC' TSEF: 'TPC Test Step EF' TSGH: 'TPC Test Step GH' MPEDch: 'Max. Power E-DCH' ULCM: 'TPC Test Step UL CM' CTFC: 'Change of TFC' DHIB: 'DC HSPA In-Band Emission'

set_timeout(*timeout: float*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:TOUT
driver.configure.tpc.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.tpc.clone()
```

Subgroups

7.2.7.1 IlpControl

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:TPC:ILPControl:MLength
CONFIGure:WCDMa:MEASurement<Instance>:TPC:ILPControl:TSEF
CONFIGure:WCDMa:MEASurement<Instance>:TPC:ILPControl:TSGH
CONFIGure:WCDMa:MEASurement<Instance>:TPC:ILPControl:TSSegment
CONFIGure:WCDMa:MEASurement<Instance>:TPC:ILPControl:AEXecution
```

class IlpControl

IlpControl commands group definition. 5 total commands, 0 Sub-groups, 5 group commands

class TsefStruct

Structure for reading output parameters. Fields:

- Length: int: numeric Number of TPC bits per test step Range: 100 to 170
- Statistics: int: numeric Number of slots at the end of test step E (F) , where the minimum (maximum) output power results are measured. Range: 1 slot to 20 slots, Unit: slots

class TsghStruct

Structure for reading output parameters. Fields:

- Length: int: numeric Number of TPC bits per test step Range: 60 to 170
- Statistics: int: numeric Number of slots at the end of test step G (H) , where the minimum (maximum) output power results are measured. Range: 1 slot to 20 slots, Unit: slots

get_aexecution() → bool

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:TPC:ILPControl:AEXecution
value: bool = driver.configure.tpc.ilpControl.get_aexecution()
```

Enables or disables automatic execution of the TPC setup for combined signal path measurements in ‘Inner Loop Power Control’ mode.

return enable: OFF | ON

get_mlength() → int

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:TPC:ILPControl:MLENght
value: int = driver.configure.tpc.ilpControl.get_mlength()
```

Query the number of slots measured in ‘Inner Loop Power Control’ mode. The value depends on the selected TPC setup and the test step settings. It can only be determined while the ‘Inner Loop Power Control’ mode is active. In other modes INV is returned.

return meas_length: decimal Range: 101 slots to 341 slots, Unit: slots

get_ts_segment() → bool

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:TPC:ILPControl:TSSegment
value: bool = driver.configure.tpc.ilpControl.get_ts_segment()
```

Enables or disables segmentation for test steps E, F, G and H. For the combined signal path scenario, use CONFIGure:WCDma:SIGN<i>:UL:TPCSet:PCONfig:TSSegment.

return enable: OFF | ON

get_tsef() → TsefStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:TPC:ILPControl:TSEF
value: TsefStruct = driver.configure.tpc.ilpControl.get_tsef()
```

Configures the inner loop power control test steps E and F. For the combined signal path scenario, use CONFIGure:WCDma:SIGN<i>:UL:TPCSet:PCONfig:TSEF.

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

get_tsgh() → TsghStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSGH
value: TsghStruct = driver.configure.tpc.ilpControl.get_tsgh()
```

Configures the inner loop power control test steps G and H. For the combined signal path scenario, use method `RsCmwWcdmaMeas.Configure.Tpc.IlpControl.tsgh`.

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

set_aexecution(*enable: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:AExecution
driver.configure.tpc.ilpControl.set_aexecution(enable = False)
```

Enables or disables automatic execution of the TPC setup for combined signal path measurements in 'Inner Loop Power Control' mode.

param enable OFF | ON

set_ts_segment(*enable: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSSegment
driver.configure.tpc.ilpControl.set_ts_segment(enable = False)
```

Enables or disables segmentation for test steps E, F, G and H. For the combined signal path scenario, use `CONFIGure:WCDMa:SIGN<i>:UL:TPCSet:PCONfig:TSSegment`.

param enable OFF | ON

set_tsef(*value: RsCmwWcdmaMeas.Implementations.Configure_.Tpc_.IlpControl.IlpControl.TsefStruct*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSEF
driver.configure.tpc.ilpControl.set_tsef(value = TsefStruct())
```

Configures the inner loop power control test steps E and F. For the combined signal path scenario, use `CONFIGure:WCDMa:SIGN<i>:UL:TPCSet:PCONfig:TSEF`.

param value see the help for `TsefStruct` structure arguments.

set_tsgh(*value: RsCmwWcdmaMeas.Implementations.Configure_.Tpc_.IlpControl.IlpControl.TsghStruct*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSGH
driver.configure.tpc.ilpControl.set_tsgh(value = TsghStruct())
```

Configures the inner loop power control test steps G and H. For the combined signal path scenario, use method `RsCmwWcdmaMeas.Configure.Tpc.IlpControl.tsgh`.

param value see the help for `TsghStruct` structure arguments.

7.2.7.2 Monitor

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:TPC:MONitor:MLENgtH
```

class Monitor

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

get_mlength() → int

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MONitor:MLENgtH
value: int = driver.configure.tpc.monitor.get_mlength()
```

Defines the number of slots to be measured in 'Monitor' mode.

return meas_length: numeric Range: 1 slot to 341 slots, Unit: slots

set_mlength(meas_length: int) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MONitor:MLENgtH
driver.configure.tpc.monitor.set_mlength(meas_length = 1)
```

Defines the number of slots to be measured in 'Monitor' mode.

param meas_length numeric Range: 1 slot to 341 slots, Unit: slots

7.2.7.3 Mpedch

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:TPC:MPEDch:MLENgtH
CONFigure:WCDMa:MEASurement<Instance>:TPC:MPEDch:AEXecution
```

class Mpedch

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

get_aexecution() → bool

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MPEDch:AEXecution
value: bool = driver.configure.tpc.mpedch.get_aexecution()
```

Enables or disables automatic execution of the TPC setup for combined signal path measurements in 'Max. Power E-DCH' mode.

return enable: OFF | ON

get_mlength() → int

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MPEDch:MLENgtH
value: int = driver.configure.tpc.mpedch.get_mlength()
```

Defines the number of slots to be measured in 'Max. Power E-DCH' mode.

return meas_length: numeric Range: 1 slot to 341 slots, Unit: slots

set_aexecution(enable: bool) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MPEDch:AEXecution
driver.configure.tpc.mpedch.set_aexecution(enable = False)
```

Enables or disables automatic execution of the TPC setup for combined signal path measurements in 'Max. Power E-DCH' mode.

param enable OFF | ON

set_mlength(meas_length: int) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MPEDch:MLENgtH
driver.configure.tpc.mpedch.set_mlength(meas_length = 1)
```

Defines the number of slots to be measured in 'Max. Power E-DCH' mode.

param meas_length numeric Range: 1 slot to 341 slots, Unit: slots

7.2.7.4 Ctfc

class Ctfc

Ctfc 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.tpc.ctfc.clone()
```

Subgroups

7.2.7.4.1 Mlength

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:TPC:CTFC:MLENgtH
```

class Mlength

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

class GetStruct

Response structure. Fields:

- **Nr_Steps**: int: numeric Number of steps to be measured per direction Range: 1 to 5
- **Meas_Length**: int: decimal Number of slots to be measured Range: 1 slot to 301 slot , Unit: slot

get() → GetStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:CTFC:MLENgtH
value: GetStruct = driver.configure.tpc.ctfc.mlength.get()
```

Specifies the number of power steps to be measured per step direction (n up steps + n down steps) . A query returns the configured number of steps and the resulting measurement length.

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

set(nr_steps: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:CTFC:MLENgtH
driver.configure.tpc.ctfc.mlength.set(nr_steps = 1)
```

Specifies the number of power steps to be measured per step direction (n up steps + n down steps) . A query returns the configured number of steps and the resulting measurement length.

param nr_steps numeric Number of steps to be measured per direction Range: 1 to 5

7.2.7.5 Ulcm

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:TPC:ULCM:MLENgtH
CONFIGure:WCDMa:MEASurement<Instance>:TPC:ULCM:AEXecution
```

class Ulcm

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

get_aexecution() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:AEXecution
value: bool = driver.configure.tpc.ulcm.get_aexecution()
```

Enables or disables automatic execution of the TPC setup for combined signal path measurements in ‘UL Compressed Mode’ mode.

return enable: OFF | ON

get_mlength() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:MLENgtH
value: int = driver.configure.tpc.ulcm.get_mlength()
```

Query the number of slots measured in ‘UL Compressed Mode’ mode. The value is fixed. It can only be determined while the ‘UL Compressed Mode’ mode is active.

return meas_length: numeric Range: 60 slots , Unit: slots

set_aexecution(enable: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:AEXecution
driver.configure.tpc.ulcm.set_aexecution(enable = False)
```


Enables or disables automatic execution of the TPC setup for combined signal path measurements in 'UL Compressed Mode' mode.

param enable OFF | ON

set_mlength(meas_length: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:MLEngth
driver.configure.tpc.ulcm.set_mlength(meas_length = 1)
```

Query the number of slots measured in 'UL Compressed Mode' mode. The value is fixed. It can only be determined while the 'UL Compressed Mode' mode is active.

param meas_length numeric Range: 60 slots , Unit: slots

7.2.7.6 Dhib

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:TPC:DHIB:MLEngth
CONFIGure:WCDMa:MEASurement<Instance>:TPC:DHIB:PATtern
CONFIGure:WCDMa:MEASurement<Instance>:TPC:DHIB:AEExecution
```

class Dhib

Dhib commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

get_aexecution() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:AEExecution
value: bool = driver.configure.tpc.dhib.get_aexecution()
```

Enables or disables automatic execution of the TPC setup for combined signal path measurements in 'In-band Emission' mode.

return enable: OFF | ON

get_mlength() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:MLEngth
value: int = driver.configure.tpc.dhib.get_mlength()
```

Defines the number of slots to be measured in 'DC HSDPA In-Band Emission' mode.

return meas_length: numeric Range: 1 to 20, Unit: slots

get_pattern() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:PATtern
value: float = driver.configure.tpc.dhib.get_pattern()
```

Specifies the pattern and in the same time also the carrier to be tested. Select the pattern 00... for the tested carrier and 11... for the other carrier.

return pattern: numeric | UD | DU UD: C1: 11... C2: 00... DU: C1: 00... C2: 11...

set_aexecution(*enable: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:AEXecution
driver.configure.tpc.dhib.set_aexecution(enable = False)
```

Enables or disables automatic execution of the TPC setup for combined signal path measurements in ‘In-band Emission’ mode.

param enable OFF | ON

set_mlength(*meas_length: int*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:MLENgtH
driver.configure.tpc.dhib.set_mlength(meas_length = 1)
```

Defines the number of slots to be measured in ‘DC HSDPA In-Band Emission’ mode.

param meas_length numeric Range: 1 to 20, Unit: slots

set_pattern(*pattern: float*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:PATtern
driver.configure.tpc.dhib.set_pattern(pattern = 1.0)
```

Specifies the pattern and in the same time also the carrier to be tested. Select the pattern 00... for the tested carrier and 11... for the other carrier.

param pattern numeric | UD | DU UD: C1: 11... C2: 00... DU: C1: 00... C2: 11...

7.2.7.7 Limit

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:MPEDch
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:CTFC
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:DHIB
```

class Limit

Limit commands group definition. 13 total commands, 2 Sub-groups, 3 group commands

class CtfcStruct

Structure for reading output parameters. Fields:

- **Power_Step_Limit**: float: numeric Symmetrical tolerance value for the power step size Range: 0 dB to 10 dB, Unit: dB
- **Calc_Beta_Factors**: bool: OFF | ON Enables or disables the automatic calculation of the expected power step size from the configured beta factors
- **Power_Step_Size**: float: numeric Expected power step size applicable if the automatic calculation from beta factors is disabled Range: 0 dB to 24 dB, Unit: dB

class MpedchStruct

Structure for reading output parameters. Fields:

- **Enable**: bool: OFF | ON Disables | enables the limit check

- **Nom_Max_Power**: float: numeric Nominal maximum UE power Range: -47 dBm to 34 dBm, Unit: dBm
- **Upper_Limit**: float: numeric Upper limit = nominal power + this value Range: 0 dB to 10 dB, Unit: dB
- **Lower_Limit**: float: numeric Lower limit = nominal power + this value Range: -10 dB to 0 dB, Unit: dB

get_ctfc() → CtfcStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:CTFC
value: CtfcStruct = driver.configure.tpc.limit.get_ctfc()
```

Configures a power step limit for the measurement mode ‘Change of TFC’.

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

get_dhib() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:DHIB
value: float = driver.configure.tpc.limit.get_dhib()
```

Defines a ‘DC HSPA In-Band Emission’ limit: upper limit for the ratio of the UE output power in one carrier to the UE output power in the other carrier.

return min_power: numeric Range: -80 dB to 0 dB, Unit: dB

get_mpedch() → MpedchStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:MPEDch
value: MpedchStruct = driver.configure.tpc.limit.get_mpedch()
```

Configures UE power limits for the measurement mode ‘Max. Power E-DCH’.

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

set_ctfc(value: RsCmwWcdmaMeas.Implementations.Configure_.Tpc_.Limit.Limit.CtfcStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:CTFC
driver.configure.tpc.limit.set_ctfc(value = CtfcStruct())
```

Configures a power step limit for the measurement mode ‘Change of TFC’.

param value see the help for CtfcStruct structure arguments.

set_dhib(min_power: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:DHIB
driver.configure.tpc.limit.set_dhib(min_power = 1.0)
```

Defines a ‘DC HSPA In-Band Emission’ limit: upper limit for the ratio of the UE output power in one carrier to the UE output power in the other carrier.

param min_power numeric Range: -80 dB to 0 dB, Unit: dB

set_mpedch(value: RsCmwWcdmaMeas.Implementations.Configure_.Tpc_.Limit.Limit.MpedchStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:MPEDch
driver.configure.tpc.limit.set_mpedch(value = MpedchStruct())
```

Configures UE power limits for the measurement mode 'Max. Power E-DCH'.

param value see the help for MpedchStruct structure arguments.

Cloning the Group

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

Subgroups

7.2.7.7.1 IlpControl

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ILPControl:MINPower
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ILPControl:PSTep
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ILPControl:EPSTep
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ILPControl:PSGroup
```

class IlpControl

IlpControl commands group definition. 8 total commands, 1 Sub-groups, 4 group commands

class EpStepStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON
- Max_Count: int: numeric Maximum allowed exceptions for sections BC, EF and GH Range: 1 to 10
- Step_1_Db: float: numeric Exceptional limit for step size 1 dB Range: 0 dB to 5 dB
- Step_2_Db: float: numeric Exceptional limit for step size 2 dB Range: 0 dB to 5 dB

class MinPowerStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Upper_Limit: float: numeric Range: -70 dBm to 34 dBm, Unit: dBm

class PsGroupStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Group_10_X_0_Db: float: numeric Limit for groups with expected step size 10 x 0 dB (algorithm 2) Range: 0 dB to 9 dB, Unit: dB
- Group_10_X_1_Dba_Lg_2: float: numeric Limit for groups with expected step size 10 x ± 1 dB + 40 x 0 dB (algorithm 2) Range: 0 dB to 9 dB, Unit: dB

- Group_10_X_1_Db: float: numeric Limit for groups with expected step size $10 \times \pm 1$ dB (algorithm 1) Range: 0 dB to 9 dB, Unit: dB
- Group_10_X_2_Db: float: numeric Limit for groups with expected step size $10 \times \pm 2$ dB (algorithm 1) Range: 0 dB to 9 dB, Unit: dB

class PstepStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Step_0_Db: float: numeric Limit for steps with expected step size 0 dB Range: 0 dB to 5 dB, Unit: dB
- Step_1_Db: float: numeric Limit for steps with expected step size ± 1 dB Range: 0 dB to 5 dB, Unit: dB
- Step_2_Db: float: numeric Limit for steps with expected step size ± 2 dB Range: 0 dB to 5 dB, Unit: dB

get_ep_step() → EpStepStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:EPSTep
value: EpStepStruct = driver.configure.tpc.limit.ilpControl.get_ep_step()
```

Defines ‘Inner Loop Power Control’ limits for exceptions and enables or disables the limit check.

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

get_min_power() → MinPowerStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MINPower
value: MinPowerStruct = driver.configure.tpc.limit.ilpControl.get_min_power()
```

Defines an ‘Inner Loop Power Control’ limit: upper limit for the minimum UE output power. Also enables or disables the limit check.

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

get_ps_group() → PsGroupStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSGroup
value: PsGroupStruct = driver.configure.tpc.limit.ilpControl.get_ps_group()
```

Defines ‘Inner Loop Power Control’ limits: upper limits for the absolute value of the power step group error, depending on the expected step size. Also enables or disables the limit check.

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

get_pstep() → PstepStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSTep
value: PstepStruct = driver.configure.tpc.limit.ilpControl.get_pstep()
```

Defines ‘Inner Loop Power Control’ limits: upper limits for the absolute value of the power step error, depending on the expected step size. Also enables or disables the limit check.

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

set_ep_step(value: RsCmwWcd-
maMeas.Implementations.Configure_.Tpc_.Limit_.IlpControl.IlpControl.EpStepStruct) →
None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:EPSTep
driver.configure.tpc.limit.ilpControl.set_ep_step(value = EpStepStruct())
```

Defines ‘Inner Loop Power Control’ limits for exceptions and enables or disables the limit check.

param value see the help for EpStepStruct structure arguments.

set_min_power(value: RsCmwWcd-
maMeas.Implementations.Configure_.Tpc_.Limit_.IlpControl.IlpControl.MinPowerStruct)
→ None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MINPower
driver.configure.tpc.limit.ilpControl.set_min_power(value = MinPowerStruct())
```

Defines an ‘Inner Loop Power Control’ limit: upper limit for the minimum UE output power. Also enables or disables the limit check.

param value see the help for MinPowerStruct structure arguments.

set_ps_group(value: RsCmwWcd-
maMeas.Implementations.Configure_.Tpc_.Limit_.IlpControl.IlpControl.PsGroupStruct) →
None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSGrouP
driver.configure.tpc.limit.ilpControl.set_ps_group(value = PsGroupStruct())
```

Defines ‘Inner Loop Power Control’ limits: upper limits for the absolute value of the power step group error, depending on the expected step size. Also enables or disables the limit check.

param value see the help for PsGroupStruct structure arguments.

set_pstep(value: RsCmwWcd-
maMeas.Implementations.Configure_.Tpc_.Limit_.IlpControl.IlpControl.PstepStruct) →
None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSTep
driver.configure.tpc.limit.ilpControl.set_pstep(value = PstepStruct())
```

Defines ‘Inner Loop Power Control’ limits: upper limits for the absolute value of the power step error, depending on the expected step size. Also enables or disables the limit check.

param value see the help for PstepStruct structure arguments.

Cloning the Group

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

Subgroups

7.2.7.7.1.1 MaxPower

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ILPControl:MAXPower:URPClass
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ILPControl:MAXPower:ACTive
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ILPControl:MAXPower:UDEfined
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ILPControl:MAXPower
```

class MaxPower

MaxPower commands group definition. 4 total commands, 0 Sub-groups, 4 group commands

class ActiveStruct

Structure for reading output parameters. Fields:

- Nominal_Max_Power: float: float Nominal maximum output power of the UE Range: -50 dBm to 34 dBm, Unit: dBm
- Upper_Limit: float: float Tolerance value for too high maximum UE power Range: 0 dB to 5 dB, Unit: dB
- Lower_Limit: float: float Tolerance value for too low maximum UE power Range: -5 dB to 0 dB, Unit: dB

class UserDefinedStruct

Structure for reading output parameters. Fields:

- Nominal_Max_Power: float: numeric Nominal maximum output power of the UE Range: -50 dBm to 34 dBm, Unit: dBm
- Upper_Limit: float: numeric Tolerance value for too high maximum UE power Range: 0 dB to 5 dB, Unit: dB
- Lower_Limit: float: numeric Tolerance value for too low maximum UE power Range: -5 dB to 0 dB, Unit: dB

class ValueStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Active_Limit: enums.ActiveLimit: USER | PC1 | PC2 | PC3 | PC3B | PC4 To use the limits defined by 3GPP, select the power class of the UE (PC1 to PC4 = power class 1, 2, 3, 3bis, 4) . To use the UE power class value reported by the UE in the capability report, see also [CMDLINK: CONFIGure:WCDMa:MEASi:TPC:LIMit:ILPControl:MAXPower:URPClass CMDLINK]. For user-defined limit values, select USER and define the limits via [CMDLINK: CONFIGure:WCDMa:MEASi:TPC:LIMit:ILPControl:MAXPower:UDEfined CMDLINK].

get_active() → ActiveStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:TPC:LIMit:ILPControl:MAXPower:ACTive
value: ActiveStruct = driver.configure.tpc.limit.ilpControl.maxPower.get_
↳active()
```

Queries the active limit values for the ‘Inner Loop Power Control’ mode. These limit values result either from the configured UE power class or from the reported UE power class or have been defined manually.

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

get_urp_class() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:TPC:LIMit:ILPControl:MAXPower:URPClass
value: bool = driver.configure.tpc.limit.ilpControl.maxPower.get_urp_class()
```

Enables or disables the use of the UE power class value reported by the UE in the capability report. This command is only relevant for combined signal path ‘Inner Loop Power Control’ measurements and only if the predefined limit sets are used.

return enable: OFF | ON

get_user_defined() → UserDefinedStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:TPC:LIMit:ILPControl:MAXPower:UDEFined
value: UserDefinedStruct = driver.configure.tpc.limit.ilpControl.maxPower.get_
↳user_defined()
```

Sets the user-defined maximum output power limits for the ‘Inner Loop Power Control’ mode. To activate the usage of this limit set, see method RsCmwWcdmaMeas.Configure.Tpc.Limit.IlpControl.MaxPower.value.

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

get_value() → ValueStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MAXPower
value: ValueStruct = driver.configure.tpc.limit.ilpControl.maxPower.get_value()
```

Enables or disables the check of the maximum UE output power limits for the ‘Inner Loop Power Control’ mode and selects the set of limit settings to be used.

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

set_urp_class(enable: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:TPC:LIMit:ILPControl:MAXPower:URPClass
driver.configure.tpc.limit.ilpControl.maxPower.set_urp_class(enable = False)
```

Enables or disables the use of the UE power class value reported by the UE in the capability report. This command is only relevant for combined signal path ‘Inner Loop Power Control’ measurements and only if

the predefined limit sets are used.

param enable OFF | ON

set_user_defined(*value*: RsCmwWcdmaMeas.Implementations.Configure_Tpc_Limit_IlpControl_MaxPower.MaxPower.UserDefinedStruct)
→ None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:TPC:LIMit:ILPControl:MAXPower:UDEFined
driver.configure.tpc.limit.ilpControl.maxPower.set_user_defined(value =
↳UserDefinedStruct())
```

Sets the user-defined maximum output power limits for the ‘Inner Loop Power Control’ mode. To activate the usage of this limit set, see method RsCmwWcdmaMeas.Configure.Tpc.Limit.IlpControl.MaxPower.value.

param value see the help for UserDefinedStruct structure arguments.

set_value(*value*: RsCmwWcdmaMeas.Implementations.Configure_Tpc_Limit_IlpControl_MaxPower.MaxPower.ValueStruct)
→ None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MAXPower
driver.configure.tpc.limit.ilpControl.maxPower.set_value(value = ValueStruct())
```

Enables or disables the check of the maximum UE output power limits for the ‘Inner Loop Power Control’ mode and selects the set of limit settings to be used.

param value see the help for ValueStruct structure arguments.

7.2.7.7.2 Ulcm

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ULCM:PA
CONFIGure:WCDMa:MEASurement<Instance>:TPC:LIMit:ULCM:PB
```

class Ulcm

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

class PaStruct

Structure for reading output parameters. Fields:

- **Initial_Pwr_Step**: float or bool: numeric | ON | OFF Symmetrical tolerance value for UE TX power in the first slot after the gap Range: 0 dB to 10 dB, Unit: dB Additional parameters: OFF | ON (disables | enables the limit)
- **Power_Step**: float or bool: numeric | ON | OFF Symmetrical tolerance value for UE TX power in a recovery period Range: 0 dB to 10 dB, Unit: dB Additional parameters: OFF | ON (disables | enables the limit)
- **Power_Step_Group**: float or bool: numeric | ON | OFF Symmetrical tolerance value for the aggregate UE TX power in the recovery period comprising the 7 rising or falling power steps after each gap Range: 0 dB to 10 dB, Unit: dB Additional parameters: OFF | ON (disables | enables the limit)

class PbStruct

Structure for reading output parameters. Fields:

- **Initial_Pwr_Step**: float or bool: numeric | ON | OFF Symmetrical tolerance value for the UE TX power in the first slot after the gap Range: 0 dB to 10 dB, Unit: dB Additional parameters: OFF | ON (disables | enables the limit)
- **Power_Step**: float or bool: numeric | ON | OFF Symmetrical tolerance value for the UE TX power in the nonCM - CM and CM - nonCM power step Range: 0 dB to 10 dB, Unit: dB Additional parameters: OFF | ON (disables | enables the limit)

get_pa() → PaStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PA
value: PaStruct = driver.configure.tpc.limit.ulcm.get_pa()
```

Configures a power step limit for the measurement mode 'UL Compressed Mode', CM pattern A.

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

get_pb() → PbStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PB
value: PbStruct = driver.configure.tpc.limit.ulcm.get_pb()
```

Configures a power step limit for the measurement mode 'UL Compressed Mode', CM pattern B.

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

set_pa(value: RsCmwWcdmaMeas.Implementations.Configure_.Tpc_.Limit_.UlcM.Ulcm.PaStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PA
driver.configure.tpc.limit.ulcm.set_pa(value = PaStruct())
```

Configures a power step limit for the measurement mode 'UL Compressed Mode', CM pattern A.

param value see the help for PaStruct structure arguments.

set_pb(value: RsCmwWcdmaMeas.Implementations.Configure_.Tpc_.Limit_.UlcM.Ulcm.PbStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PB
driver.configure.tpc.limit.ulcm.set_pb(value = PbStruct())
```

Configures a power step limit for the measurement mode 'UL Compressed Mode', CM pattern B.

param value see the help for PbStruct structure arguments.

7.2.8 Prach

SCPI Commands

```

CONFigure:WCDMa:MEASurement<Instance>:PRACH:TOUT
CONFigure:WCDMa:MEASurement<Instance>:PRACH:MPreamble
CONFigure:WCDMa:MEASurement<Instance>:PRACH:PPreamble
CONFigure:WCDMa:MEASurement<Instance>:PRACH:OFFPower
CONFigure:WCDMa:MEASurement<Instance>:PRACH:MOEXception

```

class Prach

Prach commands group definition. 30 total commands, 2 Sub-groups, 5 group commands

get_mo_exception() → bool

```

# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:MOEXception
value: bool = driver.configure.prach.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 OFF: Faulty results are rejected. ON: Results are never rejected.

get_mpreamble() → int

```

# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:MPreamble
value: int = driver.configure.prach.get_mpreamble()

```

Specifies the number of preambles to be measured.

return preambles: decimal Range: 1 to 5

get_off_power() → bool

```

# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:OFFPower
value: bool = driver.configure.prach.get_off_power()

```

Enables or disables the measurement of the off power before and after the last preamble.

return enable: OFF | ON

get_ppreamble() → int

```

# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:PPreamble
value: int = driver.configure.prach.get_ppreamble()

```

Selects the preamble used to determine the single preamble results, i.e. the ‘... vs Chip’ results and the I/Q diagram. The number of the preselected preamble must be smaller than the number of measured preambles (method RsCmwWcdmaMeas. Configure.Prach.mpreamble) .

return preamble: integer Range: 0 to 4

get_timeout() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRCh:TOUT
value: float = driver.configure.prach.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:WCDMa:MEASurement<instance>:PRCh:MOException
driver.configure.prach.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 OFF: Faulty results are rejected. ON: Results are never rejected.

set_mpreamble(preambles: int) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRCh:MPreamble
driver.configure.prach.set_mpreamble(preambles = 1)
```

Specifies the number of preambles to be measured.

param preambles decimal Range: 1 to 5

set_off_power(enable: bool) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRCh:OFFPower
driver.configure.prach.set_off_power(enable = False)
```

Enables or disables the measurement of the off power before and after the last preamble.

param enable OFF | ON

set_ppreamble(preamble: int) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRCh:PPreamble
driver.configure.prach.set_ppreamble(preamble = 1)
```

Selects the preamble used to determine the single preamble results, i.e. the ‘... vs Chip’ results and the I/Q diagram. The number of the preselected preamble must be smaller than the number of measured preambles (method RsCmwWcdmaMeas. Configure.Prach.mpreamble) .

param preamble integer Range: 0 to 4

set_timeout(*timeout: float*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:TOUT
driver.configure.prach.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.prach.clone()
```

Subgroups

7.2.8.1 Limit

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:EVMagnitude
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:MERRor
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:PERRor
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:IQOffset
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:IQIMbalance
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:CFERror
```

class Limit

Limit commands group definition. 13 total commands, 1 Sub-groups, 6 group commands

class EvMagnitudeStruct

Structure for reading output parameters. Fields:

- Rms: float or bool: numeric | ON | OFF Range: 0 % to 99 %, Unit: % Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)
- Peak: float or bool: numeric | ON | OFF Range: 0 % to 99 %, 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:

- Rms: float or bool: numeric | ON | OFF Range: 0 % to 99 %, Unit: % Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

- Peak: float or bool: numeric | ON | OFF Range: 0 % to 99 %, 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:

- Rms: float or bool: numeric | ON | OFF Range: 0 deg to 45 deg, Unit: deg Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)
- Peak: float or bool: numeric | ON | OFF Range: 0 deg to 45 deg, Unit: deg Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_cf_error() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:CFError
value: float or bool = driver.configure.prach.limit.get_cf_error()
```

Defines an upper limit for the carrier frequency error.

return frequency_error: numeric | ON | OFF Range: 0 Hz to 4000 Hz, Unit: Hz Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_ev_magnitude() → EvMagnitudeStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:EVMagnitude
value: EvMagnitudeStruct = driver.configure.prach.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:WCDMa:MEASurement<instance>:PRCh:LIMit:IQIMbalance
value: float or bool = driver.configure.prach.limit.get_iq_imbalance()
```

Defines an upper limit for the I/Q imbalance.

return iq_imbalance: numeric | ON | OFF Range: -99 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_iq_offset() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:IQOffset
value: float or bool = driver.configure.prach.limit.get_iq_offset()
```

Defines an upper limit for the I/Q origin offset.

return iq_offset: numeric | ON | OFF Range: -80 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_merror() → MerrorStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:MERRor
value: MerrorStruct = driver.configure.prach.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_perror() → PerrorStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PERror
value: PerrorStruct = driver.configure.prach.limit.get_perror()
```

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

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

set_cf_error(frequency_error: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:CFERror
driver.configure.prach.limit.set_cf_error(frequency_error = 1.0)
```

Defines an upper limit for the carrier frequency error.

param frequency_error numeric | ON | OFF Range: 0 Hz to 4000 Hz, Unit: Hz Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_ev_magnitude(value: RsCmwWcdmaMeas.Implementations.Configure_Prach_Limit.Limit.EvMagnitudeStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:EVMagnitude
driver.configure.prach.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:WCDMa:MEASurement<instance>:PRACH:LIMit:IQIMbalance
driver.configure.prach.limit.set_iq_imbalance(iq_imbalance = 1.0)
```

Defines an upper limit for the I/Q imbalance.

param iq_imbalance numeric | ON | OFF Range: -99 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_iq_offset(iq_offset: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:IQOffset
driver.configure.prach.limit.set_iq_offset(iq_offset = 1.0)
```

Defines an upper limit for the I/Q origin offset.

param iq_offset numeric | ON | OFF Range: -80 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_merror(value: RsCmwWcdmaMeas.Implementations.Configure_.Prach_.Limit.Limit.MerrorStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:MERRor
driver.configure.prach.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_perror(value: RsCmwWcdmaMeas.Implementations.Configure_.Prach_.Limit.Limit.PerrorStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PERRor
driver.configure.prach.limit.set_perror(value = PerrorStruct())
```

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

param value see the help for PerrorStruct structure arguments.

Cloning the Group

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

Subgroups

7.2.8.1.1 Pcontrol

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:PRACH:LIMit:PCONtrol:PSTep
CONFIGure:WCDMa:MEASurement<Instance>:PRACH:LIMit:PCONtrol:OLPower
CONFIGure:WCDMa:MEASurement<Instance>:PRACH:LIMit:PCONtrol:OFFPower
```

class Pcontrol

Pcontrol commands group definition. 7 total commands, 1 Sub-groups, 3 group commands

class OlPowerStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Init_Preamble_Pwr: float: numeric Initial preamble power Range: -50 dBm to 34 dBm, Unit: dBm
- Olp_Limit: float: numeric Open loop power tolerance value Range: 0 dB to 15 dB, Unit: dB

class PstepStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Preamble_Pwr_Step: float: numeric Expected preamble power step size Range: 0 dB to 15 dB, Unit: dB
- Pwr_Step_Limit: float: numeric Preamble power step tolerance value Range: 0 dB to 15 dB, Unit: dB

get_off_power() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:OFFPower
value: float or bool = driver.configure.prach.limit.pcontrol.get_off_power()
```

Defines an upper OFF power limit. Also enables or disables the limit check.

return limit: numeric | ON | OFF Range: -90 dBm to 53 dBm, Unit: dBm Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

get_ol_power() → OlPowerStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:OLPower
value: OlPowerStruct = driver.configure.prach.limit.pcontrol.get_ol_power()
```

Enables or disables the check of the open loop power limits and specifies these limits.

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

get_pstep() → PstepStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:PSTep
value: PstepStruct = driver.configure.prach.limit.pcontrol.get_pstep()
```

Enables or disables the check of the preamble power step limits and specifies these limits.

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

set_off_power(limit: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:OFFPower
driver.configure.prach.limit.pcontrol.set_off_power(limit = 1.0)
```

Defines an upper OFF power limit. Also enables or disables the limit check.

param limit numeric | ON | OFF Range: -90 dBm to 53 dBm, Unit: dBm Additional parameters: OFF | ON (disables the limit check | enables the limit check using the previous/default limit values)

set_ol_power(value: RsCmwWcdmaMeas.Implementations.Configure_.Prach_.Limit_.Pcontrol.Pcontrol.OlPowerStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:OLPower
driver.configure.prach.limit.pcontrol.set_ol_power(value = OlPowerStruct())
```

Enables or disables the check of the open loop power limits and specifies these limits.

param value see the help for OIPowerStruct structure arguments.

set_pstep(value:
RsCmwWcdmaMeas.Implementations.Configure_.Prach_.Limit_.Pcontrol.Pcontrol.PstepStruct)
 → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:PCONtrol:PSTep
driver.configure.prach.limit.pcontrol.set_pstep(value = PstepStruct())
```

Enables or disables the check of the preamble power step limits and specifies these limits.

param value see the help for PstepStruct structure arguments.

Cloning the Group

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

Subgroups

7.2.8.1.1.1 MaxPower

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:PCONtrol:MAXPower:URPClass
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:PCONtrol:MAXPower:ACTive
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:PCONtrol:MAXPower:UDEFined
CONFIGure:WCDMa:MEASurement<Instance>:PRCh:LIMit:PCONtrol:MAXPower
```

class MaxPower

MaxPower commands group definition. 4 total commands, 0 Sub-groups, 4 group commands

class ActiveStruct

Structure for reading output parameters. Fields:

- **Nominal_Max_Power**: float: float Nominal maximum output power of the UE Range: -50 dBm to 34 dBm, Unit: dBm
- **Upper_Limit**: float: float Tolerance value for too high maximum UE power Range: 0 dB to 5 dB, Unit: dB
- **Lower_Limit**: float: float Tolerance value for too low maximum UE power Range: -5 dB to 0 dB, Unit: dB

class UserDefinedStruct

Structure for reading output parameters. Fields:

- **Nominal_Max_Power**: float: numeric Nominal maximum output power of the UE Range: -50 dBm to 34 dBm, Unit: dBm
- **Upper_Limit**: float: numeric Tolerance value for too high maximum UE power Range: 0 dB to 5 dB, Unit: dB

- Lower_Limit: float: numeric Tolerance value for too low maximum UE power Range: -5 dB to 0 dB, Unit: dB

class ValueStruct

Structure for reading output parameters. Fields:

- Enable: bool: OFF | ON Disables | enables the limit check
- Active_Limit: enums.ActiveLimit: USER | PC1 | PC2 | PC3 | PC3B | PC4 To use the limits defined by 3GPP, select the power class of the UE (PC1 to PC4 = power class 1, 2, 3, 3bis, 4) . To use the UE power class value reported by the UE in the capability report, see also [CMDLINK: CONFigure:WCDMa:MEASi:PRACH:LIMit:PCONtrol:MAXPower:URPClass CMDLINK]. For user-defined limit values, select USER and define the limits via [CMDLINK: CONFigure:WCDMa:MEASi:PRACH:LIMit:PCONtrol:MAXPower:UDEFined CMDLINK].

get_active() → ActiveStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:PRACH:LIMit:PCONtrol:MAXPower:ACTive
value: ActiveStruct = driver.configure.prach.limit.pcontrol.maxPower.get_
↳active()
```

Queries the active maximum output power limit values. These limit values result either from the configured or reported UE power class or have been specified manually.

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

get_urp_class() → bool

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:PRACH:LIMit:PCONtrol:MAXPower:URPClass
value: bool = driver.configure.prach.limit.pcontrol.maxPower.get_urp_class()
```

Enables or disables the usage of the UE power class value reported by the UE in the capability report. This setting is only relevant if the combined signal path scenario is active and not relevant if user-defined limits are used instead of the predefined limit sets.

return enable: OFF | ON

get_user_defined() → UserDefinedStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:PRACH:LIMit:PCONtrol:MAXPower:UDEFined
value: UserDefinedStruct = driver.configure.prach.limit.pcontrol.maxPower.get_
↳user_defined()
```

Sets the user-defined maximum output power limits. To activate the usage of this limit set, see method RsCmwWcdmaMeas. Configure.Prach.Limit.Pcontrol.MaxPower.value.

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

get_value() → ValueStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:MAXPower
value: ValueStruct = driver.configure.prach.limit.pcontrol.maxPower.get_value()
```

Enables or disables the check of the maximum output power limits and selects the set of limit settings to be used.

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

set_urp_class(enable: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:PRACH:LIMit:PCONtrol:MAXPower:URPClass
driver.configure.prach.limit.pcontrol.maxPower.set_urp_class(enable = False)
```

Enables or disables the usage of the UE power class value reported by the UE in the capability report. This setting is only relevant if the combined signal path scenario is active and not relevant if user-defined limits are used instead of the predefined limit sets.

param enable OFF | ON

set_user_defined(value: RsCmwWcd-
maMeas.Implementations.Configure_.Prach_.Limit_.Pcontrol_.MaxPower.MaxPower.UserDefinedStruct)
→ None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:PRACH:LIMit:PCONtrol:MAXPower:UDEFined
driver.configure.prach.limit.pcontrol.maxPower.set_user_defined(value =  
↳UserDefinedStruct())
```

Sets the user-defined maximum output power limits. To activate the usage of this limit set, see method RsCmwWcdmaMeas. Configure.Prach.Limit.Pcontrol.MaxPower.value.

param value see the help for UserDefinedStruct structure arguments.

set_value(value: RsCmwWcd-
maMeas.Implementations.Configure_.Prach_.Limit_.Pcontrol_.MaxPower.MaxPower.ValueStruct)
→ None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:MAXPower
driver.configure.prach.limit.pcontrol.maxPower.set_value(value = ValueStruct())
```

Enables or disables the check of the maximum output power limits and selects the set of limit settings to be used.

param value see the help for ValueStruct structure arguments.

7.2.8.2 Result

SCPI Commands

```

CONFigure:WCDMa:MEASurement<Instance>:PRACH:RESult:UEPower
CONFigure:WCDMa:MEASurement<Instance>:PRACH:RESult:PSTeps
CONFigure:WCDMa:MEASurement<Instance>:PRACH:RESult:FERRor
CONFigure:WCDMa:MEASurement<Instance>:PRACH:RESult:ALL
CONFigure:WCDMa:MEASurement<Instance>:PRACH:RESult:PERRor
CONFigure:WCDMa:MEASurement<Instance>:PRACH:RESult:EVMagnitude
CONFigure:WCDMa:MEASurement<Instance>:PRACH:RESult:MERRor
CONFigure:WCDMa:MEASurement<Instance>:PRACH:RESult:IQ

```

class Result

Result commands group definition. 12 total commands, 1 Sub-groups, 8 group commands

class AllStruct

Structure for reading output parameters. Fields:

- Enable_Ue_Power: bool: OFF | ON UE power OFF: Do not evaluate results, hide the view ON: Evaluate results and show the view
- Enable_Pow_Steps: bool: OFF | ON Power steps
- Enable_Freq_Error: bool: OFF | ON Frequency error
- Enable_Evm: bool: OFF | ON Error vector magnitude
- Enable_Mag_Error: bool: OFF | ON Magnitude error
- Enable_Phase_Err: bool: OFF | ON Phase error
- Enable_Ue_Pchip: bool: OFF | ON UE power vs. chip
- Enable_Evm_Chip: bool: OFF | ON EVM vs. chip
- Enable_Merr_Chip: bool: OFF | ON Magnitude error vs. chip
- Enable_Ph_Err_Chip: bool: OFF | ON Phase error vs. chip
- Enable_Iq: bool: OFF | ON I/Q constellation diagram

get_all() → AllStruct

```

# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:RESult[:ALL]
value: AllStruct = driver.configure.prach.result.get_all()

```

Enables or disables the evaluation of results and shows or hides the views in the PRACH measurement. This command combines all other CONFigure:WCDMa:MEAS<i>:PRACH:RESult... commands.

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

get_ev_magnitude() → bool

```

# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:RESult:EVMagnitude
value: bool = driver.configure.prach.result.get_ev_magnitude()

```

Enables or disables the evaluation of results and shows or hides the error vector magnitude view in the PRACH measurement.

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

get_freq_error() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:FERRor
value: bool = driver.configure.prach.result.get_freq_error()
```

Enables or disables the evaluation of results and shows or hides the frequency error view in the PRACH measurement.

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

get_iq() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:IQ
value: bool = driver.configure.prach.result.get_iq()
```

Enables or disables the evaluation of results and shows or hides the I/Q constellation diagram view in the PRACH measurement.

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

get_merror() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:MERRor
value: bool = driver.configure.prach.result.get_merror()
```

Enables or disables the evaluation of results and shows or hides the magnitude error view in the PRACH measurement.

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

get_perror() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:PERRor
value: bool = driver.configure.prach.result.get_perror()
```

Enables or disables the evaluation of results and shows or hides the phase error view in the PRACH measurement.

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

get_psteps() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:PSTeps
value: bool = driver.configure.prach.result.get_psteps()
```

Enables or disables the evaluation of results and shows or hides the power steps view in the PRACH measurement.

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

get_ue_power() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:UEPower
value: bool = driver.configure.prach.result.get_ue_power()
```

Enables or disables the evaluation of results and shows or hides the UE power view in the PRACH measurement.

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

set_all(value: RsCmwWcdmaMeas.Implementations.Configure_.Prach_Result.Result.AllStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult[:ALL]
driver.configure.prach.result.set_all(value = AllStruct())
```

Enables or disables the evaluation of results and shows or hides the views in the PRACH measurement. This command combines all other CONFIGure:WCDMa:MEAS<i>:PRACH:RESult... commands.

param value see the help for AllStruct structure arguments.

set_ev_magnitude(enable_evm: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:EVMagnitude
driver.configure.prach.result.set_ev_magnitude(enable_evm = False)
```

Enables or disables the evaluation of results and shows or hides the error vector magnitude view in the PRACH measurement.

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

set_freq_error(enable_freq_error: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:FERRor
driver.configure.prach.result.set_freq_error(enable_freq_error = False)
```

Enables or disables the evaluation of results and shows or hides the frequency error view in the PRACH measurement.

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

set_iq(enable_iq: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:IQ
driver.configure.prach.result.set_iq(enable_iq = False)
```

Enables or disables the evaluation of results and shows or hides the I/Q constellation diagram view in the PRACH measurement.

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

set_merror(*enable_mag_error: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:MERRor
driver.configure.prach.result.set_merror(enable_mag_error = False)
```

Enables or disables the evaluation of results and shows or hides the magnitude error view in the PRACH measurement.

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

set_perror(*enable_phase_err: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:PERRor
driver.configure.prach.result.set_perror(enable_phase_err = False)
```

Enables or disables the evaluation of results and shows or hides the phase error view in the PRACH measurement.

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

set_psteps(*enable_pow_steps: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:PSTeps
driver.configure.prach.result.set_psteps(enable_pow_steps = False)
```

Enables or disables the evaluation of results and shows or hides the power steps view in the PRACH measurement.

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

set_ue_power(*enable_ue_power: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:UEPower
driver.configure.prach.result.set_ue_power(enable_ue_power = False)
```

Enables or disables the evaluation of results and shows or hides the UE power view in the PRACH measurement.

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

Cloning the Group

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

Subgroups

7.2.8.2.1 Chip

SCPI Commands

```
CONFIGure:WCDMa:MEASurement<Instance>:PRACH:RESult:CHIP:UEPower
CONFIGure:WCDMa:MEASurement<Instance>:PRACH:RESult:CHIP:PERRor
CONFIGure:WCDMa:MEASurement<Instance>:PRACH:RESult:CHIP:MERRor
CONFIGure:WCDMa:MEASurement<Instance>:PRACH:RESult:CHIP:EVM
```

class Chip

Chip commands group definition. 4 total commands, 0 Sub-groups, 4 group commands

get_evm() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:EVM
value: bool = driver.configure.prach.result.chip.get_evm()
```

Enables or disables the evaluation of results and shows or hides the EVM vs. chip view in the PRACH measurement.

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

get_merror() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:MERRor
value: bool = driver.configure.prach.result.chip.get_merror()
```

Enables or disables the evaluation of results and shows or hides the magnitude error vs. chip view in the PRACH measurement.

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

get_perror() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:PERRor
value: bool = driver.configure.prach.result.chip.get_perror()
```

Enables or disables the evaluation of results and shows or hides the phase error vs. chip view in the PRACH measurement.

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

get_ue_power() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:UEPower
value: bool = driver.configure.prach.result.chip.get_ue_power()
```

Enables or disables the evaluation of results and shows or hides the UE power vs. chip view in the PRACH measurement.

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

set_evm(enable_evm_chip: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:EVM
driver.configure.prach.result.chip.set_evm(enable_evm_chip = False)
```

Enables or disables the evaluation of results and shows or hides the EVM vs. chip view in the PRACH measurement.

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

set_merror(enable_merr_chip: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:MERRor
driver.configure.prach.result.chip.set_merror(enable_merr_chip = False)
```

Enables or disables the evaluation of results and shows or hides the magnitude error vs. chip view in the PRACH measurement.

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

set_perror(enable_ph_err_chip: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:PERRor
driver.configure.prach.result.chip.set_perror(enable_ph_err_chip = False)
```

Enables or disables the evaluation of results and shows or hides the phase error vs. chip view in the PRACH measurement.

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

set_ue_power(enable_ue_pchip: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:UEPower
driver.configure.prach.result.chip.set_ue_power(enable_ue_pchip = False)
```

Enables or disables the evaluation of results and shows or hides the UE power vs. chip view in the PRACH measurement.

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

7.2.9 OoSync

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:OOSync:AADPchlevel
CONFigure:WCDMa:MEASurement<Instance>:OOSync:TOUT
```

class OoSync

OoSync commands group definition. 5 total commands, 1 Sub-groups, 2 group commands

get_aa_dpch_level() → bool

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:AADPchlevel
value: bool = driver.configure.ooSync.get_aa_dpch_level()
```

Enables or disables automatic activation of DPCH level sequence, provided by WCDMA signaling application. With auto execution, the sequence is activated by starting the measurement.

return enable: OFF | ON

get_timeout() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:TOUT
value: float = driver.configure.ooSync.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_aa_dpch_level(enable: bool) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:AADPchlevel
driver.configure.ooSync.set_aa_dpch_level(enable = False)
```

Enables or disables automatic activation of DPCH level sequence, provided by WCDMA signaling application. With auto execution, the sequence is activated by starting the measurement.

param enable OFF | ON

set_timeout(timeout: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:TOUT
driver.configure.ooSync.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.ooSync.clone()
```

Subgroups

7.2.9.1 Limit

SCPI Commands

```
CONFigure:WCDMa:MEASurement<Instance>:OOSync:LIMit:PONupper
CONFigure:WCDMa:MEASurement<Instance>:OOSync:LIMit:POFFupper
CONFigure:WCDMa:MEASurement<Instance>:OOSync:LIMit:THReshold
```

class Limit

Limit commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

get_poff_upper() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:LIMit:POFFupper
value: float = driver.configure.ooSync.limit.get_poff_upper()
```

Specifies the transmitted power of the UE below which the UE's transmitter is considered to be off.

return po_ulimit: numeric Range: -90 dBm to 53 dBm

get_pon_upper() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:LIMit:PONupper
value: float = driver.configure.ooSync.limit.get_pon_upper()
```

Specifies the transmitted power of the UE above which the UE's transmitter is considered to be on.

return pon_lower: numeric Range: -70 dBm to 34 dBm

get_threshold() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:LIMit:THReshold
value: float = driver.configure.ooSync.limit.get_threshold()
```

Specifies the reliability of results for ‘RX Level Strategy’ ‘Max A off F Max’. If the UE transmitter is expected to be on and the UE power is below the limit, results are not reliable.

return threshold_level: numeric Range: -65 dB to 0 dB

set_poff_upper(po_ulimit: float) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:OOSync:LIMit:POFFupper
driver.configure.ooSync.limit.set_poff_upper(po_ulimit = 1.0)
```

Specifies the transmitted power of the UE below which the UE’s transmitter is considered to be off.

param po_ulimit numeric Range: -90 dBm to 53 dBm

set_pon_upper(pon_lower: float) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:OOSync:LIMit:PONupper
driver.configure.ooSync.limit.set_pon_upper(pon_lower = 1.0)
```

Specifies the transmitted power of the UE above which the UE’s transmitter is considered to be on.

param pon_lower numeric Range: -70 dBm to 34 dBm

set_threshold(threshold_level: float) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:OOSync:LIMit:THReshold
driver.configure.ooSync.limit.set_threshold(threshold_level = 1.0)
```

Specifies the reliability of results for ‘RX Level Strategy’ ‘Max A off F Max’. If the UE transmitter is expected to be on and the UE power is below the limit, results are not reliable.

param threshold_level numeric Range: -65 dB to 0 dB

7.2.10 OlpControl

SCPI Commands

```
CONFIGure:WCDma:MEASurement<Instance>:OLPControl:TOUT
CONFIGure:WCDma:MEASurement<Instance>:OLPControl:MOEXception
CONFIGure:WCDma:MEASurement<Instance>:OLPControl:LIMit
```

class OlpControl

OlpControl commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

get_limit() → float

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:OLPControl:LIMit
value: float = driver.configure.olpControl.get_limit()
```

Sets the maximum deviation at any carrier regarding the expected nominal UE TX power.

return olp_limit: numeric Upper limit for DPCCH preamble power Range: 0 dB to 15 dB, Unit: dB

get_mo_exception() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:MOEXception
value: bool = driver.configure.olpControl.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 OFF: faulty results are rejected ON: results are never rejected

get_timeout() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:TOUT
value: float = driver.configure.olpControl.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_limit(olp_limit: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:LIMit
driver.configure.olpControl.set_limit(olp_limit = 1.0)
```

Sets the maximum deviation at any carrier regarding the expected nominal UE TX power.

param olp_limit numeric Upper limit for DPCCH preamble power Range: 0 dB to 15 dB, Unit: dB

set_mo_exception(meas_on_exception: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:MOEXception
driver.configure.olpControl.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 OFF: faulty results are rejected ON: results are never rejected

set_timeout(timeout: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:TOUT
driver.configure.olpControl.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

7.3 Trigger

class Trigger

Trigger commands group definition. 36 total commands, 5 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:WCDMa:MEASurement<Instance>:MEValuation:DElay
TRIGger:WCDMa:MEASurement<Instance>:MEValuation:MGAP
TRIGger:WCDMa:MEASurement<Instance>:MEValuation:SOURce
TRIGger:WCDMa:MEASurement<Instance>:MEValuation:THReshold
TRIGger:WCDMa:MEASurement<Instance>:MEValuation:SLOPe
TRIGger:WCDMa:MEASurement<Instance>:MEValuation:TOUT
```

class MultiEval

MultiEval commands group definition. 8 total commands, 2 Sub-groups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:WCDMa: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. A delay is useful if the trigger event and the uplink DPCH slot border are not synchronous. A measurement starts always at an uplink DPCH slot border. Triggering a measurement at another time can yield a synchronization error. For internal trigger sources aligned to the downlink DPCH, an additional delay of 1024 chips is automatically applied. It corresponds to the assumed delay between downlink and uplink slot. This setting has no influence on free run measurements.

return delay: numeric Range: -666.7E-6 s to 0.24 s, Unit: s

get_mgap() → float

```
# SCPI: TRIGger:WCDMa: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 minimum_gap: numeric Range: 0 s to 0.01 s, Unit: s

get_slope() → RsCmwWcdmaMeas.enums.SignalSlope

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEvaluation:SLOPe
value: enums.SignalSlope = 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:WCDMa: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 source: string ‘Free Run (Standard)’: Free run (standard synchronization) ‘Free Run (Fast Sync)’: Free run (fast synchronization) ‘IF Power’: Power trigger (normal synchronization) ‘IF Power (Sync)’: Power trigger (extended synchronization)

get_threshold() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEvaluation:THReshold
value: float = driver.trigger.multiEval.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return level: numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

get_timeout() → float

```
# SCPI: TRIGger:WCDMa: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 timeout: numeric | ON | OFF Range: 0.01 s to 10 s, Unit: s Additional OFF | ON disables/enables the timeout

set_delay(*delay: float*) → None

```
# SCPI: TRIGger:WCDMa: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. A delay is useful if the trigger event and the uplink DPCH slot border are not synchronous. A measurement starts always at an uplink DPCH slot border. Triggering a measurement at another time can yield a synchronization error. For internal trigger sources aligned to the downlink DPCH, an additional delay of 1024 chips is automatically applied. It corresponds to the assumed delay between downlink and uplink slot. This setting has no influence on free run measurements.

param delay numeric Range: -666.7E-6 s to 0.24 s, Unit: s

set_mgap(*minimum_gap: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:MGAP
driver.trigger.multiEval.set_mgap(minimum_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 minimum_gap numeric Range: 0 s to 0.01 s, Unit: s

set_slope(*slope: RsCmwWcdmaMeas.enums.SignalSlope*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:SLOPe
driver.trigger.multiEval.set_slope(slope = enums.SignalSlope.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(*source: str*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:SOURce
driver.trigger.multiEval.set_source(source = '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 source string ‘Free Run (Standard)’: Free run (standard synchronization) ‘Free Run (Fast Sync)’: Free run (fast synchronization) ‘IF Power’: Power trigger (normal synchronization) ‘IF Power (Sync)’: Power trigger (extended synchronization)

set_threshold(*level: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:THReshold
driver.trigger.multiEval.set_threshold(level = 1.0)
```

Defines the trigger threshold for power trigger sources.

param level numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

set_timeout(*timeout: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:TOUT
driver.trigger.multiEval.set_timeout(timeout = 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 timeout numeric | ON | OFF Range: 0.01 s to 10 s, Unit: s Additional OFF | ON disables/enables the timeout

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:WCDMa: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:WCDMa: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 RsCmwWcdmaMeas.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:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODE
```

class ListPy

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

get_mode() → RsCmwWcdmaMeas.enums.Mode

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:LIST:MODE
value: enums.Mode = driver.trigger.multiEval.listPy.get_mode()
```

Specifies the trigger mode for list mode measurements. For configuration of retrigger flags, see method RsCmwWcdmaMeas. Configure.MultiEval.ListPy.Segment.Setup.set.

return mode: ONCE | SEGMENT ONCE: A trigger event is only required to start the measurement. As a result, the entire range of segments to be measured is captured without additional trigger event. The retrigger flags of the segments are ignored. SEGMENT: The retrigger flag of each segment is evaluated. It defines whether the measurement waits for a trigger event before capturing the segment, or not.

set_mode(mode: RsCmwWcdmaMeas.enums.Mode) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:LIST:MODE
driver.trigger.multiEval.listPy.set_mode(mode = enums.Mode.ONCE)
```

Specifies the trigger mode for list mode measurements. For configuration of retrigger flags, see method RsCmwWcdmaMeas. Configure.MultiEval.ListPy.Segment.Setup.set.

param mode ONCE | SEGMENT ONCE: A trigger event is only required to start the measurement. As a result, the entire range of segments to be measured is captured without additional trigger event. The retrigger flags of the segments are ignored. SEGMENT: The retrigger flag of each segment is evaluated. It defines whether the measurement waits for a trigger event before capturing the segment, or not.

7.3.2 Tpc

SCPI Commands

```
TRIGger:WCDMa:MEASurement<Instance>:TPC:DElay
TRIGger:WCDMa:MEASurement<Instance>:TPC:MGAP
TRIGger:WCDMa:MEASurement<Instance>:TPC:SOURce
TRIGger:WCDMa:MEASurement<Instance>:TPC:THReshold
TRIGger:WCDMa:MEASurement<Instance>:TPC:SLOPe
TRIGger:WCDMa:MEASurement<Instance>:TPC:TOUT
```

class Tpc

Tpc commands group definition. 7 total commands, 1 Sub-groups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:DElay
value: float = driver.trigger.tpc.get_delay()
```

Defines a time delaying the start of the measurement relative to the trigger event. The delay is useful if the trigger event and the uplink DPCH slot border are not synchronous. A measurement starts always at an uplink DPCH slot border. Triggering a measurement at another time yields a synchronization error.

For internal trigger sources aligned to the downlink DPCH, an additional delay of 1024 chips is automatically applied. It corresponds to the assumed delay between downlink and uplink slot. This setting has no influence on ‘Free Run’ measurements.

return delay: numeric Range: -666.7E-6 s to 0.24 s, Unit: s

get_mgap() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:MGAP
value: float = driver.trigger.tpc.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 minimum_gap: numeric Range: 0 s to 0.01 s, Unit: s

get_slope() → RsCmwWcdmaMeas.enums.SignalSlope

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:SLOPe
value: enums.SignalSlope = driver.trigger.tpc.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:WCDMa:MEASurement<instance>:TPC:SOURce
value: str = driver.trigger.tpc.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 source: string ‘Free Run (Standard)’: Free run (standard synchronization) ‘Free Run (Fast Sync)’: Free run (fast synchronization) ‘IF Power’: Power trigger (normal synchronization) ‘IF Power (Sync)’: Power trigger (extended synchronization)

get_threshold() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:THReshold
value: float = driver.trigger.tpc.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return threshold: numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

get_timeout() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:TOUT
value: float or bool = driver.trigger.tpc.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 timeout: numeric | ON | OFF Range: 0.01 s to 10 s, Unit: s Additional parameters: OFF | ON (disables | enables the timeout)

set_delay(*delay: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:DElay
driver.trigger.tpc.set_delay(delay = 1.0)
```

Defines a time delaying the start of the measurement relative to the trigger event. The delay is useful if the trigger event and the uplink DPCH slot border are not synchronous. A measurement starts always at an uplink DPCH slot border. Triggering a measurement at another time yields a synchronization error. For internal trigger sources aligned to the downlink DPCH, an additional delay of 1024 chips is automatically applied. It corresponds to the assumed delay between downlink and uplink slot. This setting has no influence on ‘Free Run’ measurements.

param delay numeric Range: -666.7E-6 s to 0.24 s, Unit: s

set_mgap(*minimum_gap: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:MGAP
driver.trigger.tpc.set_mgap(minimum_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 minimum_gap numeric Range: 0 s to 0.01 s, Unit: s

set_slope(*slope: RsCmwWcdmaMeas.enums.SignalSlope*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:SLOPe
driver.trigger.tpc.set_slope(slope = enums.SignalSlope.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(*source: str*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:SOURce
driver.trigger.tpc.set_source(source = '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 source string ‘Free Run (Standard)’: Free run (standard synchronization) ‘Free Run (Fast Sync)’: Free run (fast synchronization) ‘IF Power’: Power trigger (normal synchronization) ‘IF Power (Sync)’: Power trigger (extended synchronization)

set_threshold(*threshold: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:THReshold
driver.trigger.tpc.set_threshold(threshold = 1.0)
```

Defines the trigger threshold for power trigger sources.

param threshold numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

set_timeout(*timeout: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:TOUT
driver.trigger.tpc.set_timeout(timeout = 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 timeout numeric|ON|OFF Range: 0.01 s to 10 s, Unit: s Additional parameters:
OFF | ON (disables | enables the timeout)

Cloning the Group

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

Subgroups

7.3.2.1 Catalog

SCPI Commands

```
TRIGger:WCDMa:MEASurement<Instance>:TPC:CATalog:SOURce
```

class Catalog

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

get_source() → List[str]

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:CATalog:SOURce
value: List[str] = driver.trigger.tpc.catalog.get_source()
```

Lists all trigger source values that can be set using method RsCmwWcdmaMeas.Trigger.Tpc.source.

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

7.3.3 Prach

SCPI Commands

```
TRIGger:WCDMa:MEASurement<Instance>:PRACH:DELay
TRIGger:WCDMa:MEASurement<Instance>:PRACH:MGAP
TRIGger:WCDMa:MEASurement<Instance>:PRACH:SOURce
TRIGger:WCDMa:MEASurement<Instance>:PRACH:THReshold
TRIGger:WCDMa:MEASurement<Instance>:PRACH:SLOPe
TRIGger:WCDMa:MEASurement<Instance>:PRACH:TOUT
```

class Prach

Prach commands group definition. 7 total commands, 1 Sub-groups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:DELay
value: float = driver.trigger.prach.get_delay()
```

Defines a time delaying the start of the measurement relative to the trigger event.

return delay: numeric Range: -666.7E-6 s to 0.24 s, Unit: s

get_mgap() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:MGAP
value: float = driver.trigger.prach.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 minimum_gap: numeric Range: 0 s to 0.01 s, Unit: s

get_slope() → RsCmwWcdmaMeas.enums.SignalSlope

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:SLOPe
value: enums.SignalSlope = driver.trigger.prach.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:WCDMa:MEASurement<instance>:PRACH:SOURce
value: str = driver.trigger.prach.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 source: string 'IF Power (Sync) ': Power trigger (extended synchronization)

get_threshold() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRCh:THReshold
value: float = driver.trigger.prach.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return level: numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

get_timeout() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRCh:TOUT
value: float or bool = driver.trigger.prach.get_timeout()
```

Selects the maximum time that the R&S CMW waits for a trigger event before it stops the measurement in remote control mode or indicates a trigger timeout in manual operation mode.

return timeout: numeric | ON | OFF Range: 0.01 s to 60 s, Unit: s Additional parameters: OFF | ON (disables | enables the timeout)

set_delay(delay: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRCh:DElay
driver.trigger.prach.set_delay(delay = 1.0)
```

Defines a time delaying the start of the measurement relative to the trigger event.

param delay numeric Range: -666.7E-6 s to 0.24 s, Unit: s

set_mgap(minimum_gap: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRCh:MGAP
driver.trigger.prach.set_mgap(minimum_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 minimum_gap numeric Range: 0 s to 0.01 s, Unit: s

set_slope(slope: RsCmwWcdmaMeas.enums.SignalSlope) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRCh:SLOPe
driver.trigger.prach.set_slope(slope = enums.SignalSlope.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(source: str) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRCh:SOURce
driver.trigger.prach.set_source(source = '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 source string 'IF Power (Sync) ': Power trigger (extended synchronization)

set_threshold(level: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:THReshold
driver.trigger.prach.set_threshold(level = 1.0)
```

Defines the trigger threshold for power trigger sources.

param level numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

set_timeout(timeout: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:TOUT
driver.trigger.prach.set_timeout(timeout = 1.0)
```

Selects the maximum time that the R&S CMW waits for a trigger event before it stops the measurement in remote control mode or indicates a trigger timeout in manual operation mode.

param timeout numeric | ON | OFF Range: 0.01 s to 60 s, Unit: s Additional parameters: OFF | ON (disables | enables the timeout)

Cloning the Group

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

Subgroups

7.3.3.1 Catalog

SCPI Commands

```
TRIGger:WCDMa:MEASurement<Instance>:PRACH:CATalog:SOURce
```

class Catalog

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

get_source() → List[str]

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:CATalog:SOURce
value: List[str] = driver.trigger.prach.catalog.get_source()
```

Lists all trigger source values that can be set using method RsCmwWcdmaMeas.Trigger.Prach.source.

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

7.3.4 OoSync

SCPI Commands

```
TRIGger:WCDma:MEASurement<Instance>:OOSync:DElay
TRIGger:WCDma:MEASurement<Instance>:OOSync:MGAP
TRIGger:WCDma:MEASurement<Instance>:OOSync:SOURce
TRIGger:WCDma:MEASurement<Instance>:OOSync:THReshold
TRIGger:WCDma:MEASurement<Instance>:OOSync:SLOPe
TRIGger:WCDma:MEASurement<Instance>:OOSync:TOUT
```

class OoSync

OoSync commands group definition. 7 total commands, 1 Sub-groups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:OOSync:DElay
value: float = driver.trigger.ooSync.get_delay()
```

Defines a time delaying the start of the measurement relative to the trigger event. The delay is useful if the trigger event and the uplink DPCH slot border are not synchronous. A measurement starts always at an uplink DPCH slot border. Triggering a measurement at another time can yield a synchronization error. For internal trigger sources aligned to the downlink DPCH, an additional delay of 1024 chips is automatically applied. It corresponds to the assumed delay between downlink and uplink slot. This setting has no influence on 'Free Run' measurements.

return delay: numeric Range: -666.7E-6 s to 0.24 s, Unit: s

get_mgap() → float

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:OOSync:MGAP
value: float = driver.trigger.ooSync.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 minimum_gap: numeric Range: 0 s to 0.01 s, Unit: s

get_slope() → RsCmwWcdmaMeas.enums.SignalSlope

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:OOSync:SLOPe
value: enums.SignalSlope = driver.trigger.ooSync.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:WCDma:MEASurement<instance>:OOSync:SOURce
value: str = driver.trigger.ooSync.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 source: string ‘Free Run (Standard)’: Free run (standard synchronization) ‘Free Run (Fast Sync)’: Free run (fast synchronization) ‘IF Power’: Power trigger (normal synchronization) ‘IF Power (Sync)’: Power trigger (extended synchronization)

get_threshold() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:THReshold
value: float = driver.trigger.ooSync.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return level: numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

get_timeout() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:TOUT
value: float or bool = driver.trigger.ooSync.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 timeout: numeric | ON | OFF Range: 0.01 s to 60 s, Unit: s Additional parameters: OFF | ON (disables | enables the timeout)

set_delay(delay: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:DElay
driver.trigger.ooSync.set_delay(delay = 1.0)
```

Defines a time delaying the start of the measurement relative to the trigger event. The delay is useful if the trigger event and the uplink DPCH slot border are not synchronous. A measurement starts always at an uplink DPCH slot border. Triggering a measurement at another time can yield a synchronization error. For internal trigger sources aligned to the downlink DPCH, an additional delay of 1024 chips is automatically applied. It corresponds to the assumed delay between downlink and uplink slot. This setting has no influence on ‘Free Run’ measurements.

param delay numeric Range: -666.7E-6 s to 0.24 s, Unit: s

set_mgap(minimum_gap: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:MGAP
driver.trigger.ooSync.set_mgap(minimum_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 minimum_gap numeric Range: 0 s to 0.01 s, Unit: s

set_slope(slope: RsCmwWcdmaMeas.enums.SignalSlope) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:SLOPe
driver.trigger.ooSync.set_slope(slope = enums.SignalSlope.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(source: str) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:SOURce
driver.trigger.ooSync.set_source(source = '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 source string ‘Free Run (Standard)’: Free run (standard synchronization) ‘Free Run (Fast Sync)’: Free run (fast synchronization) ‘IF Power’: Power trigger (normal synchronization) ‘IF Power (Sync)’: Power trigger (extended synchronization)

set_threshold(level: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:THReshold
driver.trigger.ooSync.set_threshold(level = 1.0)
```

Defines the trigger threshold for power trigger sources.

param level numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

set_timeout(timeout: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:TOUT
driver.trigger.ooSync.set_timeout(timeout = 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 timeout numeric | ON | OFF Range: 0.01 s to 60 s, Unit: s Additional parameters: OFF | ON (disables | enables the timeout)

Cloning the Group

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

Subgroups

7.3.4.1 Catalog

SCPI Commands

```
TRIGger:WCDMa:MEASurement<Instance>:OOSync:CATalog:SOURce
```

class Catalog

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

get_source() → List[str]

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:CATalog:SOURce
value: List[str] = driver.trigger.ooSync.catalog.get_source()
```

Lists all trigger source values that can be set using method RsCmwWcdmaMeas.Trigger.OoSync.source.

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

7.3.5 OlpControl

SCPI Commands

```
TRIGger:WCDMa:MEASurement<Instance>:OLPControl:DElay
TRIGger:WCDMa:MEASurement<Instance>:OLPControl:MGAP
TRIGger:WCDMa:MEASurement<Instance>:OLPControl:SOURce
TRIGger:WCDMa:MEASurement<Instance>:OLPControl:THReshold
TRIGger:WCDMa:MEASurement<Instance>:OLPControl:SLOPe
TRIGger:WCDMa:MEASurement<Instance>:OLPControl:TOUT
```

class OlpControl

OlpControl commands group definition. 7 total commands, 1 Sub-groups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:DElay
value: float = driver.trigger.olpControl.get_delay()
```

Defines a time delaying the start of the measurement relative to the trigger event. The delay is useful if the trigger event and the uplink DPCH slot border are not synchronous. A measurement starts always at an uplink DPCH slot border. Triggering a measurement at another time can yield a synchronization error. For internal trigger sources aligned to the downlink DPCH, an additional delay of 1024 chips is automatically applied. It corresponds to the assumed delay between downlink and uplink slot. This setting has no influence on 'Free Run' measurements.

return delay: numeric Range: -666.7E-6 s to 0.24 s, Unit: s

get_mgap() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:MGAP
value: float = driver.trigger.olpControl.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 minimum_gap: numeric Range: 0 s to 0.01 s, Unit: s

get_slope() → RsCmwWcdmaMeas.enums.SignalSlope

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:SLOPe
value: enums.SignalSlope = driver.trigger.olpControl.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:WCDMa:MEASurement<instance>:OLPControl:SOURce
value: str = driver.trigger.olpControl.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 source: string ‘Free Run (Standard)’: Free run (standard synchronization) ‘Free Run (Fast Sync)’: Free run (fast synchronization) ‘IF Power’: Power trigger (normal synchronization) ‘IF Power (Sync)’: Power trigger (extended synchronization)

get_threshold() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:THReshold
value: float = driver.trigger.olpControl.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return level: numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

get_timeout() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:TOUT
value: float or bool = driver.trigger.olpControl.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 timeout: numeric | ON | OFF Range: 0.01 s to 60 s, Unit: s Additional parameters: OFF | ON (disables | enables the timeout)

set_delay(delay: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:DElay
driver.trigger.olpControl.set_delay(delay = 1.0)
```

Defines a time delaying the start of the measurement relative to the trigger event. The delay is useful if the trigger event and the uplink DPCH slot border are not synchronous. A measurement starts always at an uplink DPCH slot border. Triggering a measurement at another time can yield a synchronization error. For internal trigger sources aligned to the downlink DPCH, an additional delay of 1024 chips is automatically applied. It corresponds to the assumed delay between downlink and uplink slot. This setting has no influence on 'Free Run' measurements.

param delay numeric Range: -666.7E-6 s to 0.24 s, Unit: s

set_mgap(*minimum_gap*: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:MGAP
driver.trigger.olpControl.set_mgap(minimum_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 minimum_gap numeric Range: 0 s to 0.01 s, Unit: s

set_slope(*slope*: RsCmwWcdmaMeas.enums.SignalSlope) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:SLOPe
driver.trigger.olpControl.set_slope(slope = enums.SignalSlope.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(*source*: str) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:SOURce
driver.trigger.olpControl.set_source(source = '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 source string 'Free Run (Standard) ': Free run (standard synchronization) 'Free Run (Fast Sync) ': Free run (fast synchronization) 'IF Power': Power trigger (normal synchronization) 'IF Power (Sync) ': Power trigger (extended synchronization)

set_threshold(*level*: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:THReshold
driver.trigger.olpControl.set_threshold(level = 1.0)
```

Defines the trigger threshold for power trigger sources.

param level numeric Range: -47 dB to 0 dB, Unit: dB (full scale, i.e. relative to reference level minus external attenuation)

set_timeout(*timeout: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:TOUT
driver.trigger.olpControl.set_timeout(timeout = 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 timeout numeric|ON|OFF Range: 0.01 s to 60 s, Unit: s Additional parameters:
OFF | ON (disables | enables the timeout)

Cloning the Group

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

Subgroups

7.3.5.1 Catalog

SCPI Commands

```
TRIGger:WCDMa:MEASurement<Instance>:OLPControl:CATalog:SOURce
```

class Catalog

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

get_source() → List[str]

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:CATalog:SOURce
value: List[str] = driver.trigger.olpControl.catalog.get_source()
```

Lists all trigger source values that can be set using method RsCmwWcdmaMeas.Trigger.OlpControl.source.

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

7.4 MultiEval

SCPI Commands

```
STOP:WCDMa:MEASurement<Instance>:MEValuation
ABORt:WCDMa:MEASurement<Instance>:MEValuation
INITiate:WCDMa:MEASurement<Instance>:MEValuation
```

class MultiEval

MultiEval commands group definition. 529 total commands, 8 Sub-groups, 3 group commands

abort() → None

```
# SCPI: ABORt:WCDMa: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:WCDMa: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 RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

initiate() → None

```
# SCPI: INITiate:WCDMa: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.

(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.

initiate_with_opc() → None

```

# SCPI: INITiate:WCDMa: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 RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

stop() → None

```

# SCPI: STOP:WCDMa: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.
- 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:WCDMa:MEASurement<instance>:MEValuation
driver.multiEval.stop_with_opc()

```

(continues on next page)

(continued from previous page)

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 RsCmwWcdmaMeas.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:WCDMa:MEASurement<Instance>:MEValuation:STATe
```

class State

State commands group definition. 2 total commands, 1 Sub-groups, 1 group commands

fetch() → RsCmwWcdmaMeas.enums.ResourceState

```
# SCPI: FETCh:WCDMa: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 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:WCDMa: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:WCDMa: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. 61 total commands, 9 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 Phd

class Phd

Phd 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.phd.clone()
```

Subgroups

7.4.2.1.1 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:PHD:CURRENT
READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:PHD:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACe:PHD:CURRENT
value: List[float] = driver.multiEval.trace.phd.current.fetch()
```

Returns the values of the phase discontinuity traces for up to 120 slots. One value per measured slot is returned (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

INTRO_CMD_HELP: The meaning of the value depends on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) :

- For full-slot measurements, each value indicates the phase discontinuity at the boundary between a slot and the previous slot. As there is no previous slot for slot 0, the first returned phase discontinuity value equals NCAP.
- For half-slot measurements, each value indicates the phase discontinuity at the boundary between the first and second half-slot of a slot. This value can be measured for all slots, including slot 0.

See also ‘Detailed Views: Phase Discontinuity’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_disc: float One value per measured slot Range: -180 deg to 180 deg, Unit: deg

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:PHD:CURRent
value: List[float] = driver.multiEval.trace.phd.current.read()
```

Returns the values of the phase discontinuity traces for up to 120 slots. One value per measured slot is returned (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

INTRO_CMD_HELP: The meaning of the value depends on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) :

- For full-slot measurements, each value indicates the phase discontinuity at the boundary between a slot and the previous slot. As there is no previous slot for slot 0, the first returned phase discontinuity value equals NCAP.
- For half-slot measurements, each value indicates the phase discontinuity at the boundary between the first and second half-slot of a slot. This value can be measured for all slots, including slot 0.

See also ‘Detailed Views: Phase Discontinuity’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_disc: float One value per measured slot Range: -180 deg to 180 deg, Unit: deg

7.4.2.2 CdeMonitor

class CdeMonitor

CdeMonitor 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.multiEval.trace.cdeMonitor.clone()
```

Subgroups

7.4.2.2.1 Qsignal

class Qsignal

Qsignal 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.cdeMonitor.qsignal.clone()
```

Subgroups

7.4.2.2.1.1 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:CDEMonitor:QSIGnal:CURRent
READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:CDEMonitor:QSIGnal:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:CDEMonitor:QSIGnal:CURRent
value: List[float] = driver.multiEval.trace.cdeMonitor.qsignal.current.fetch()
```

Returns the values of the code domain error traces of the code domain monitor. The results of the I-Signal and Q-Signal traces can be retrieved. See also ‘Detailed Views: CD Monitor’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return qsignal: float One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) . Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:CDEMonitor:QSIGnal:CURRent
value: List[float] = driver.multiEval.trace.cdeMonitor.qsignal.current.read()
```

Returns the values of the code domain error traces of the code domain monitor. The results of the I-Signal and Q-Signal traces can be retrieved. See also ‘Detailed Views: CD Monitor’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return qsignal: float One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) . Range: -100 dB to 0 dB, Unit: dB

7.4.2.2.2 Isignal

class Isignal

Isignal 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.cdeMonitor.isignal.clone()
```

Subgroups

7.4.2.2.2.1 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:CDEMonitor:ISIGnal:CURRent
READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:CDEMonitor:ISIGnal:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:CDEMonitor:ISIGnal:CURRent
value: List[float] = driver.multiEval.trace.cdeMonitor.isignal.current.fetch()
```

Returns the values of the code domain error traces of the code domain monitor. The results of the I-Signal and Q-Signal traces can be retrieved. See also ‘Detailed Views: CD Monitor’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return isignal: float One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) . Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:CDEMonitor:ISIGnal:CURRent
value: List[float] = driver.multiEval.trace.cdeMonitor.isignal.current.read()
```

Returns the values of the code domain error traces of the code domain monitor. The results of the I-Signal and Q-Signal traces can be retrieved. See also ‘Detailed Views: CD Monitor’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return isignal: float One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) . Range: -100 dB to 0 dB, Unit: dB

7.4.2.3 CdpMonitor

class CdpMonitor

CdpMonitor 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.multiEval.trace.cdpMonitor.clone()
```

Subgroups

7.4.2.3.1 Qsignal

class Qsignal

Qsignal 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.cdpMonitor.qsignal.clone()
```

Subgroups

7.4.2.3.1.1 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:CDPMonitor:QSIGnal:CURRent
READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:CDPMonitor:QSIGnal:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪ :MEvaluation:TRACe:CDPMonitor:QSIGnal:CURRent
value: List[float] = driver.multiEval.trace.cdpMonitor.qsignal.current.fetch()
```

Returns the values of the code domain power traces of the code domain monitor. The results of the I-Signal and Q-Signal traces can be retrieved. See also ‘Detailed Views: CD Monitor’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return qsignal: float One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) . Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>
↪:MEVALUATION:TRACE:CDPMONITOR:QSIGNAL:CURRENT
value: List[float] = driver.multiEval.trace.cdpMonitor.qsignal.current.read()
```

Returns the values of the code domain power traces of the code domain monitor. The results of the I-Signal and Q-Signal traces can be retrieved. See also ‘Detailed Views: CD Monitor’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return qsignal: float One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) . Range: -100 dB to 0 dB, Unit: dB

7.4.2.3.2 Isignal

class Isignal

Isignal 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.cdpMonitor.isignal.clone()
```

Subgroups

7.4.2.3.2.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:TRACE:CDPMONITOR:ISIGNAL:CURRENT
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:TRACE:CDPMONITOR:ISIGNAL:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>
↪:MEVALUATION:TRACE:CDPMONITOR:ISIGNAL:CURRENT
value: List[float] = driver.multiEval.trace.cdpMonitor.isignal.current.fetch()
```

Returns the values of the code domain power traces of the code domain monitor. The results of the I-Signal and Q-Signal traces can be retrieved. See also ‘Detailed Views: CD Monitor’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return isignal: float One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) . Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>
↪:MEvaluation:TRACe:CDPMonitor:ISIGnal:CURRent
value: List[float] = driver.multiEval.trace.cdpMonitor.isignal.current.read()
```

Returns the values of the code domain power traces of the code domain monitor. The results of the I-Signal and Q-Signal traces can be retrieved. See also ‘Detailed Views: CD Monitor’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return isignal: float One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) . Range: -100 dB to 0 dB, Unit: dB

7.4.2.4 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.multiEval.trace.carrier.repcap_carrier_get()
driver.multiEval.trace.carrier.repcap_carrier_set(repcap.Carrier.Nr1)
```

class Carrier

Carrier commands group definition. 1 total commands, 1 Sub-groups, 0 group commands Repeated Capability: Carrier, default value after init: Carrier.Nr1

Cloning the Group

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

Subgroups

7.4.2.4.1 Perror

class Perror

Perror 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.multiEval.trace.carrier.perror.clone()
```

Subgroups

7.4.2.4.1.1 Rms

class Rms

Rms 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.multiEval.trace.carrier.perror.rms.clone()
```

Subgroups

7.4.2.4.1.2 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:CARRier<Carrier>:PERRor:RMS:CURRent
```

class Current

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

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:CARRier<carrier>
↳:PERRor[:RMS]:CURRent
value: List[float] = driver.multiEval.trace.carrier.perror.rms.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS phase error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return phase_error: No help available

7.4.2.5 Emask

class Emask

Emask commands group definition. 30 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.emask.clone()
```

Subgroups

7.4.2.5.1 MfLeft

class MfLeft

MfLeft 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.emask.mfLeft.clone()
```

Subgroups

7.4.2.5.1.1 Average

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEvaluation:TRACe:EMASk:MFLeft:AVERage
FETCh:WCDma:MEASurement<Instance>:MEvaluation:TRACe:EMASk:MFLeft:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLeft:AVERage
value: List[float] = driver.multiEval.trace.emask.mfLeft.average.fetch()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mleft: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are:

Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:TRACE:EMASK:MFLIFT:AVERAGE
value: List[float] = driver.multiEval.trace.emask.mfLeft.average.read()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mleft: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.1.2 Current

SCPI Commands

```
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:TRACE:EMASK:MFLIFT:CURRENT
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:TRACE:EMASK:MFLIFT:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:TRACE:EMASK:MFLIFT:CURRENT
value: List[float] = driver.multiEval.trace.emask.mfLeft.current.fetch()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mleft: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are:

Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:TRACE:EMASK:MFL:left:CURRENT
value: List[float] = driver.multiEval.trace.emask.mfLeft.current.read()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mleft: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.1.3 Maximum

SCPI Commands

```
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:TRACE:EMASK:MFL:left:MAXimum
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:TRACE:EMASK:MFL:left:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:TRACE:EMASK:MFL:left:MAXimum
value: List[float] = driver.multiEval.trace.emask.mfLeft.maximum.fetch()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mleft: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are:

Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLeft:MAXimum
value: List[float] = driver.multiEval.trace.emask.mfLeft.maximum.read()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mleft: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.2 MfRight

class MfRight

MfRight 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.emask.mfRight.clone()
```

Subgroups

7.4.2.5.2.1 Average

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:MEvaluation:TRACe:EMASk:MFRight:AVERage
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:TRACe:EMASk:MFRight:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:EMASk:MFRight:AVERAge
value: List[float] = driver.multiEval.trace.emask.mfRight.average.fetch()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mrigh: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFRight:AVERAge
value: List[float] = driver.multiEval.trace.emask.mfRight.average.read()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mrigh: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.2.2 Current

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:TRACe:EMASk:MFRight:CURRent
FETCh:WCDma:MEASurement<Instance>:MEValuation:TRACe:EMASk:MFRight:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>
→ :MEValuation:TRACe:EMASk:MFRight:CURRent
value: List[float] = driver.multiEval.trace.emask.mfRight.current.fetch()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mrigh: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:TRACe:EMASk:MFRight:CURRent
value: List[float] = driver.multiEval.trace.emask.mfRight.current.read()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mrigh: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.2.3 Maximum

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:TRACe:EMASk:MFRight:MAXimum
FETCh:WCDma:MEASurement<Instance>:MEValuation:TRACe:EMASk:MFRight:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>
→ :MEValuation:TRACe:EMASk:MFRight:MAXimum
value: List[float] = driver.multiEval.trace.emask.mfRight.maximum.fetch()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mrigh: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:TRACe:EMASk:MFRight:MAXimum
value: List[float] = driver.multiEval.trace.emask.mfRight.maximum.read()
```

Returns the values of the spectrum emission 1 MHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms MFLeft and MFRight) . The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_1_mrigh: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 89 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -11970 kHz to -4050 kHz from the center carrier frequency Right section: 4050 kHz to 11970 kHz from the center carrier frequency Dual carrier in uplink: n = 144 values correspond to test points that are separated by 90 kHz. The covered frequency ranges are: Left section: -19440 kHz to -6570 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Right section: 6570 kHz to 19440 kHz from the center frequency of both carriers Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.3 HkfLeft

class HkfLeft

HkfLeft 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.emask.hkfLeft.clone()
```

Subgroups

7.4.2.5.3.1 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFLeft:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFLeft:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
→ :MEValuation:TRACe:EMASk:HKFLeft:AVERage
value: List[float] = driver.multiEval.trace.emask.hkfLeft.average.fetch()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute) . See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_100_kleft: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:HKFLeft:AVERage
value: List[float] = driver.multiEval.trace.emask.hkfLeft.average.read()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends

on the limit line H mode (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute) .
See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_100_kleft: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.3.2 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFLeft:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFLeft:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:HKFLeft:CURRent
value: List[float] = driver.multiEval.trace.emask.hkfLeft.current.fetch()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute) . See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_100_kleft: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:HKFLeft:CURRent
value: List[float] = driver.multiEval.trace.emask.hkfLeft.current.read()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute) . See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_100_kleft: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.3.3 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFLeft:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFLeft:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:HKFLeft:MAXimum
value: List[float] = driver.multiEval.trace.emask.hkfLeft.maximum.fetch()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute) . See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_100_kleft: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:HKFLeft:MAXimum
value: List[float] = driver.multiEval.trace.emask.hkfLeft.maximum.read()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute) . See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_100_kleft: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.4 HkfRight

class HkfRight

HkfRight 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.emask.hkfRight.clone()
```

Subgroups

7.4.2.5.4.1 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFRight:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFRight:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:HKFRight:AVERage
value: List[float] = driver.multiEval.trace.emask.hkfRight.average.fetch()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute) . See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_100_kright: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:HKFRight:AVERage
value: List[float] = driver.multiEval.trace.emask.hkfRight.average.read()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends

on the limit line H mode (see method `RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute`) .
See also ‘Detailed Views: Spectrum Emission Mask’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

return `emask_100_kright`: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.4.2 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFRight:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFRight:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:HKFRight:CURRent
value: List[float] = driver.multiEval.trace.emask.hkfRight.current.fetch()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method `RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute`) .
See also ‘Detailed Views: Spectrum Emission Mask’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

return `emask_100_kright`: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:HKFRight:CURRent
value: List[float] = driver.multiEval.trace.emask.hkfRight.current.read()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method `RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute`) .
See also ‘Detailed Views: Spectrum Emission Mask’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

return emask_100_kright: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.4.3 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFRight:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:HKFRight:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:HKFRight:MAXimum
value: List[float] = driver.multiEval.trace.emask.hkfRight.maximum.fetch()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute) . See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_100_kright: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:HKFRight:MAXimum
value: List[float] = driver.multiEval.trace.emask.hkfRight.maximum.read()
```

Returns the values of the spectrum emission 100 kHz traces. The left section and the right section of each trace are retrieved by separate commands (distinguished by the terms HKFLeft and HKFRight) . The results of the current, average and maximum traces can be retrieved. The covered frequency range depends on the limit line H mode (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.Emask.absolute) . See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_100_kright: float These values correspond to test points that are separated by 30 kHz. The covered frequency ranges are: Left section, line H mode B/C: -12450 kHz to -3570 kHz/-2670 kHz from the carrier Right section, line H mode B/C: 3570 kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

kHz/2670 kHz to 12450 kHz from the carrier Line H mode A is not used for 100 kHz traces (NCAPs returned) Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.5 Kfilter

class Kfilter

Kfilter 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.emask.kfilter.clone()
```

Subgroups

7.4.2.5.5.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:KFILter:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:KFILter:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:KFILter:CURRent
value: List[float] = driver.multiEval.trace.emask.kfilter.current.fetch()
```

Returns the values of the spectrum emission 30 kHz traces. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_30_k: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 1665 values correspond to test points that are separated by 15 kHz and cover the frequency range between -12480 kHz and 12480 kHz from the center carrier frequency. Dual carrier in uplink: n = 2665 values correspond to test points that are separated by 15 kHz. The results cover the frequency range between -19980 kHz and 19980 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:KFILter:CURRent
value: List[float] = driver.multiEval.trace.emask.kfilter.current.read()
```

Returns the values of the spectrum emission 30 kHz traces. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_30_k: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 1665 values correspond to test points that are separated by 15 kHz and cover the frequency range between -12480 kHz and 12480 kHz from the center carrier frequency. Dual carrier in uplink: n = 2665 values correspond to test points that are separated by 15 kHz. The results cover the frequency range between -19980 kHz and 19980 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.5.2 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:KFILter:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EMASk:KFILter:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:KFILter:AVERage
value: List[float] = driver.multiEval.trace.emask.kfilter.average.fetch()
```

Returns the values of the spectrum emission 30 kHz traces. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_30_k: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 1665 values correspond to test points that are separated by 15 kHz and cover the frequency range between -12480 kHz and 12480 kHz from the center carrier frequency. Dual carrier in uplink: n = 2665 values correspond to test points that are separated by 15 kHz. The results cover the frequency range between -19980 kHz and 19980 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:KFILter:AVERage
value: List[float] = driver.multiEval.trace.emask.kfilter.average.read()
```

Returns the values of the spectrum emission 30 kHz traces. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_30_k: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: n = 1665 values correspond to test points that are separated by 15 kHz and cover the frequency range between -12480

kHz and 12480 kHz from the center carrier frequency. Dual carrier in uplink: $n = 2665$ values correspond to test points that are separated by 15 kHz. The results cover the frequency range between -19980 kHz and 19980 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Range: -100 dB to 0 dB, Unit: dB

7.4.2.5.5.3 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EMASk:KFILter:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EMASk:KFILter:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↳:MEvaluation:TRACe:EMASk:KFILter:MAXimum
value: List[float] = driver.multiEval.trace.emask.kfilter.maximum.fetch()
```

Returns the values of the spectrum emission 30 kHz traces. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_30_k: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: $n = 1665$ values correspond to test points that are separated by 15 kHz and cover the frequency range between -12480 kHz and 12480 kHz from the center carrier frequency. Dual carrier in uplink: $n = 2665$ values correspond to test points that are separated by 15 kHz. The results cover the frequency range between -19980 kHz and 19980 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Range: -100 dB to 0 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:KFILter:MAXimum
value: List[float] = driver.multiEval.trace.emask.kfilter.maximum.read()
```

Returns the values of the spectrum emission 30 kHz traces. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Spectrum Emission Mask’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_30_k: float Comma-separated list of values, the covered frequency range differs for single and dual uplink carrier: Single carrier: $n = 1665$ values correspond to test points that are separated by 15 kHz and cover the frequency range between -12480 kHz and 12480 kHz from the center carrier frequency. Dual carrier in uplink: $n = 2665$ values correspond to test points that are separated by 15 kHz. The results cover the frequency range between -19980 kHz and 19980 kHz from the center frequency of both carriers, e.g. from $f = (fC2 - fC1) / 2$. Range: -100 dB to 0 dB, Unit: dB

7.4.2.6 EvMagnitude

class EvMagnitude

EvMagnitude commands group definition. 6 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.evMagnitude.clone()
```

Subgroups

7.4.2.6.1 Chip

class Chip

Chip 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.chip.clone()
```

Subgroups

7.4.2.6.1.1 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:CHIP:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:CHIP:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪ :MEvaluation:TRACe:EVMagnitude:CHIP:MAXimum
value: List[float] = driver.multiEval.trace.evMagnitude.chip.maximum.fetch()
```

Returns the values of the RMS EVM vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas.Configure. MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_chip: float Range: 0 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:EVMagnitude:CHIP:MAXimum
value: List[float] = driver.multiEval.trace.evMagnitude.chip.maximum.read()
```

Returns the values of the RMS EVM vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas.Configure. MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_chip: float Range: 0 % to 100 %, Unit: %

7.4.2.6.1.2 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:CHIP:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EVMagnitude:CHIP:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:EVMagnitude:CHIP:AVERage
value: List[float] = driver.multiEval.trace.evMagnitude.chip.average.fetch()
```

Returns the values of the RMS EVM vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas.Configure. MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_chip: float Range: 0 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:EVMagnitude:CHIP:AVERage
value: List[float] = driver.multiEval.trace.evMagnitude.chip.average.read()
```

Returns the values of the RMS EVM vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas.Configure. MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_chip: float Range: 0 % to 100 %, Unit: %

7.4.2.6.1.3 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EVMagnitude:CHIP:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:EVMagnitude:CHIP:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EVMagnitude:CHIP:CURRent
value: List[float] = driver.multiEval.trace.evMagnitude.chip.current.fetch()
```

Returns the values of the RMS EVM vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas.Configure. MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_chip: float Range: 0 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EVMagnitude:CHIP:CURRent
value: List[float] = driver.multiEval.trace.evMagnitude.chip.current.read()
```

Returns the values of the RMS EVM vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas.Configure. MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_chip: float Range: 0 % to 100 %, Unit: %

7.4.2.7 Merror

class Merror

Merror commands group definition. 6 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.merror.clone()
```

Subgroups

7.4.2.7.1 Chip

class Chip

Chip 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.chip.clone()
```

Subgroups

7.4.2.7.1.1 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:TRACe:MERRor:CHIP:MAXimum
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:MERRor:CHIP:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:MAXimum
value: List[float] = driver.multiEval.trace.merror.chip.maximum.fetch()
```

Returns the values of the magnitude error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_chip: float Range: -100 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:MAXimum
value: List[float] = driver.multiEval.trace.merror.chip.maximum.read()
```


Returns the values of the magnitude error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_chip: float Range: -100 % to 100 %, Unit: %

7.4.2.7.1.2 Average

SCPI Commands

```

FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:MERRor:CHIP:AVERage
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:MERRor:CHIP:AVERage

```

class Average

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

fetch() → List[float]

```

# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:AVERage
value: List[float] = driver.multiEval.trace.merror.chip.average.fetch()

```

Returns the values of the magnitude error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_chip: float Range: -100 % to 100 %, Unit: %

read() → List[float]

```

# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:AVERage
value: List[float] = driver.multiEval.trace.merror.chip.average.read()

```

Returns the values of the magnitude error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_chip: float Range: -100 % to 100 %, Unit: %

7.4.2.7.1.3 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:TRACe:MERRor:CHIP:CURRent
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:MERRor:CHIP:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:CURRent
value: List[float] = driver.multiEval.trace.merror.chip.current.fetch()
```

Returns the values of the magnitude error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_chip: float Range: -100 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:CURRent
value: List[float] = driver.multiEval.trace.merror.chip.current.read()
```

Returns the values of the magnitude error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_chip: float Range: -100 % to 100 %, Unit: %

7.4.2.8 Perror

class Perror

Perror commands group definition. 6 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.perror.clone()
```

Subgroups

7.4.2.8.1 Chip

class Chip

Chip 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.chip.clone()
```

Subgroups

7.4.2.8.1.1 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:TRACe:PERRor:CHIP:MAXimum
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:PERRor:CHIP:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:MAXimum
value: List[float] = driver.multiEval.trace.perror.chip.maximum.fetch()
```

Returns the values of the RMS phase error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_chip: float Range: -180 deg to 180 deg, Unit: deg

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:MAXimum
value: List[float] = driver.multiEval.trace.perror.chip.maximum.read()
```

Returns the values of the RMS phase error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_chip: float Range: -180 deg to 180 deg, Unit: deg

7.4.2.8.1.2 Average

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:MEValuation:TRACe:PERRor:CHIP:AVERage
READ:WCDma:MEASurement<Instance>:MEValuation:TRACe:PERRor:CHIP:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:AVERage
value: List[float] = driver.multiEval.trace.perror.chip.average.fetch()
```

Returns the values of the RMS phase error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_chip: float Range: -180 deg to 180 deg, Unit: deg

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:AVERage
value: List[float] = driver.multiEval.trace.perror.chip.average.read()
```

Returns the values of the RMS phase error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_chip: float Range: -180 deg to 180 deg, Unit: deg

7.4.2.8.1.3 Current

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:MEValuation:TRACe:PERRor:CHIP:CURREnt
READ:WCDma:MEASurement<Instance>:MEValuation:TRACe:PERRor:CHIP:CURREnt
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:CURREnt
value: List[float] = driver.multiEval.trace.perror.chip.current.fetch()
```

Returns the values of the RMS phase error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_chip: float Range: -180 deg to 180 deg, Unit: deg

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:CURRent
value: List[float] = driver.multiEval.trace.perror.chip.current.read()
```

Returns the values of the RMS phase error vs. chip traces, measured in the preselected slot (see method RsCmwWcdmaMeas. Configure.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_chip: float Range: -180 deg to 180 deg, Unit: deg

7.4.2.9 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

7.4.2.9.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:TRACe:IQ:CURRent
FETCh:WCDMa: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:WCDMA: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 located on the path between two constellation points.

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

read() → ResultData

```
# SCPI: READ:WCDMA: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 located on the path between two constellation points.

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

7.4.3 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.multiEval.carrier.repcap_carrier_get()
driver.multiEval.carrier.repcap_carrier_set(repcap.Carrier.Nr1)
```

class Carrier

Carrier commands group definition. 261 total commands, 5 Sub-groups, 0 group commands Repeated Capability: Carrier, default value after init: Carrier.Nr1

Cloning the Group

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

Subgroups

7.4.3.1 Trace

class Trace

Trace commands group definition. 215 total commands, 9 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.3.1.1 UePower

class UePower

UePower 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.carrier.trace.uePower.clone()
```

Subgroups

7.4.3.1.1.1 StandardDev

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:UEPower:SDEViation
FETCH:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:UEPower:SDEViation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:UEPower:SDEViation
value: List[float] = driver.multiEval.carrier.trace.uePower.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return ue_power: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:UEPOWER:SDEViation
value: List[float] = driver.multiEval.carrier.trace.uePower.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return ue_power: No help available

7.4.3.1.2 Minimum

SCPI Commands

```
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:UEPOWER:MINIMUM
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:UEPOWER:MINIMUM
```

class Minimum

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:UEPOWER:MINIMUM
value: List[float] = driver.multiEval.carrier.trace.uePower.minimum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return ue_power: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]


```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACe:UEPower:MINimum
value: List[float] = driver.multiEval.carrier.trace.uePower.minimum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return ue_power: No help available

7.4.3.1.1.3 Maximum

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACe:UEPower:MAXimum
FETCh:WCDma:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACe:UEPower:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACe:UEPower:MAXimum
value: List[float] = driver.multiEval.carrier.trace.uePower.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return ue_power: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:UEPower:MAXimum
value: List[float] = driver.multiEval.carrier.trace.uePower.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return ue_power: No help available

7.4.3.1.1.4 Average

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:UEPower:AVERage
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:UEPower:AVERage
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:UEPower:AVERage
value: List[float] = driver.multiEval.carrier.trace.uePower.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return ue_power: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:UEPower:AVERAge
value: List[float] = driver.multiEval.carrier.trace.uePower.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return ue_power: No help available

7.4.3.1.1.5 Current

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:UEPower:CURRent
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:UEPower:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:UEPower:CURRent
value: List[float] = driver.multiEval.carrier.trace.uePower.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return ue_power: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:UEPower:CURRent
value: List[float] = driver.multiEval.carrier.trace.uePower.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return ue_power: No help available

7.4.3.1.2 EvMagnitude

class EvMagnitude

EvMagnitude 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.carrier.trace.evMagnitude.clone()
```

Subgroups

7.4.3.1.2.1 Rms

class Rms

Rms commands group definition. 8 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.3.1.2.2 Sdeviaton

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:RMS:SDEViaton
```

class Sdeviaton

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

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:SDEViaton
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.rms.sdeviaton.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS EVM traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return evm: No help available

7.4.3.1.2.3 StandardDev

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:RMS:SDEViaton
```

class StandardDev

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:SDEViaton
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.rms.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS EVM traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement

length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return evm: No help available

7.4.3.1.2.4 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:RMS:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:RMS:MAXimum
```

class Maximum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:MAXimum
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.rms.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS EVM traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return evm: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:MAXimum
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.rms.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS EVM traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`) . The number of results depends on the measurement

length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return evm: No help available

7.4.3.1.2.5 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:RMS:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:RMS:AVERage
```

class Average

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:AVERage
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.rms.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS EVM traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return evm: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:AVERage
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.rms.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS EVM traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`) . The number of results depends on the measurement

length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return evm: No help available

7.4.3.1.2.6 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:RMS:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:RMS:CURRent
```

class Current

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:CURRent
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.rms.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS EVM traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return evm: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:CURRent
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.rms.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS EVM traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`) . The number of results depends on the measurement

length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return evm: No help available

7.4.3.1.2.7 Peak

class Peak

Peak commands group definition. 8 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.3.1.2.8 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:EVMagnitude:PEAK:MAXimum
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:EVMagnitude:PEAK:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:MAXimum
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.peak.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak EVM traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return evm: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:MAXimum
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.peak.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak EVM traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return evm: No help available

7.4.3.1.2.9 StandardDev

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:EVMagnitude:PEAK:SDEVIation
```

class StandardDev

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.peak.
↳standardDev.fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak EVM traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return evm: No help available

7.4.3.1.2.10 Average

SCPI Commands

```

FETCH:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:PEAK:AVERage
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:PEAK:AVERage

```

class Average

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```

# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:AVERage
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.peak.average.
↳fetch(carrier = repcap.Carrier.Default)

```

Returns the values of the peak EVM traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return evm: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```

# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:AVERage
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.peak.average.
↳read(carrier = repcap.Carrier.Default)

```

Returns the values of the peak EVM traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return evm: No help available

7.4.3.1.2.11 Sdeviaton

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:PEAK:SDEVIaton
```

class Sdeviaton

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

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:SDEVIaton
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.peak.sdeviaton.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak EVM traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return evm: No help available

7.4.3.1.2.12 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:PEAK:CURREnt
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:EVMagnitude:PEAK:CURREnt
```

class Current

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:CURREnt
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.peak.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak EVM traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return evm: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:CURRent
value: List[float] = driver.multiEval.carrier.trace.evMagnitude.peak.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak EVM traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return evm: No help available

7.4.3.1.3 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.carrier.trace.merror.clone()
```

Subgroups

7.4.3.1.3.1 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.carrier.trace.merror.rms.clone()
```

Subgroups

7.4.3.1.3.2 StandardDev

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:MERRor[:RMS]:SDEVIation
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:MERRor[:RMS]:SDEVIation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:MERRor[:RMS]:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.merror.rms.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS magnitude error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return magnitude_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:MERRor[:RMS]:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.merror.rms.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS magnitude error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return magnitude_error: No help available

7.4.3.1.3.3 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:MERRor:RMS:MAXimum
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:MERRor:RMS:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:MERRor[:RMS]:MAXimum
value: List[float] = driver.multiEval.carrier.trace.merror.rms.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS magnitude error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return magnitude_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:MERRor[:RMS]:MAXimum
value: List[float] = driver.multiEval.carrier.trace.merror.rms.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS magnitude error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return magnitude_error: No help available

7.4.3.1.3.4 Average

SCPI Commands

```

FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:MERRor:RMS:AVERage
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:MERRor:RMS:AVERage

```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```

# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:MERRor[:RMS]:AVERage
value: List[float] = driver.multiEval.carrier.trace.merror.rms.average.
↪fetch(carrier = repcap.Carrier.Default)

```

Returns the values of the RMS magnitude error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return magnitude_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```

# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:MERRor[:RMS]:AVERage
value: List[float] = driver.multiEval.carrier.trace.merror.rms.average.
↪read(carrier = repcap.Carrier.Default)

```

Returns the values of the RMS magnitude error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return magnitude_error: No help available

7.4.3.1.3.5 Current

SCPI Commands

```

FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:MERROR:RMS:CURRENT
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:MERROR:RMS:CURRENT

```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```

# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:MERROR[:RMS]:CURRENT
value: List[float] = driver.multiEval.carrier.trace.merror.rms.current.
↳ fetch(carrier = repcap.Carrier.Default)

```

Returns the values of the RMS magnitude error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return magnitude_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```

# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:MERROR[:RMS]:CURRENT
value: List[float] = driver.multiEval.carrier.trace.merror.rms.current.
↳ read(carrier = repcap.Carrier.Default)

```

Returns the values of the RMS magnitude error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return magnitude_error: No help available

7.4.3.1.3.6 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.carrier.trace.merror.peak.clone()
```

Subgroups

7.4.3.1.3.7 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
↳:TRACe:MERRor:PEAK:SDEVIation
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
↳:TRACe:MERRor:PEAK:SDEVIation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACe:MERRor:PEAK:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.merror.peak.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak magnitude error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return magnitude_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACe:MERRor:PEAK:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.merror.peak.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak magnitude error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`. modulation). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return magnitude_error: No help available

7.4.3.1.3.8 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:MERRor:PEAK:MAXimum
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:MERRor:PEAK:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:MERRor:PEAK:MAXimum
value: List[float] = driver.multiEval.carrier.trace.merror.peak.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak magnitude error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`. modulation). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return magnitude_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:MERRor:PEAK:MAXimum
value: List[float] = driver.multiEval.carrier.trace.merror.peak.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak magnitude error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`. modulation). The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return magnitude_error: No help available

7.4.3.1.3.9 Average

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:MERROR:PEAK:AVERAGE
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:MERROR:PEAK:AVERAGE
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR:PEAK:AVERAGE
value: List[float] = driver.multiEval.carrier.trace.merror.peak.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak magnitude error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return magnitude_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR:PEAK:AVERAGE
value: List[float] = driver.multiEval.carrier.trace.merror.peak.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak magnitude error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return magnitude_error: No help available

7.4.3.1.3.10 Current

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:MERROR:PEAK:CURRENt
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:MERROR:PEAK:CURRENt
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR:PEAK:CURRENt
value: List[float] = driver.multiEval.carrier.trace.merror.peak.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak magnitude error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return magnitude_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR:PEAK:CURRENt
value: List[float] = driver.multiEval.carrier.trace.merror.peak.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak magnitude error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return magnitude_error: No help available

7.4.3.1.4 Perror

class Perror

Perror commands group definition. 15 total commands, 2 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.3.1.4.1 Rms

class Rms

Rms 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.carrier.trace.perror.rms.clone()
```

Subgroups

7.4.3.1.4.2 StandardDev

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:PERRor:RMS:SDEViation
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:PERRor:RMS:SDEViation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:PERRor[:RMS]:SDEViation
value: List[float] = driver.multiEval.carrier.trace.perror.rms.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS phase error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return phase_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:PERRor[:RMS]:SDEviation
value: List[float] = driver.multiEval.carrier.trace.perror.rms.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS phase error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return phase_error: No help available

7.4.3.1.4.3 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:PERRor:RMS:MAXimum
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:PERRor:RMS:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:PERRor[:RMS]:MAXimum
value: List[float] = driver.multiEval.carrier.trace.perror.rms.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS phase error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return phase_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:PERRor[:RMS]:MAXimum
value: List[float] = driver.multiEval.carrier.trace.perror.rms.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS phase error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return phase_error: No help available

7.4.3.1.4.4 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:PERRor:RMS:AVERage
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:PERRor:RMS:AVERage
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:PERRor[:RMS]:AVERage
value: List[float] = driver.multiEval.carrier.trace.perror.rms.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS phase error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return phase_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:PERRor[:RMS]:AVERage
value: List[float] = driver.multiEval.carrier.trace.perror.rms.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS phase error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return phase_error: No help available

7.4.3.1.4.5 Current

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:PERRor:RMS:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:PERRor[:RMS]:CURRent
value: List[float] = driver.multiEval.carrier.trace.perror.rms.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS phase error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return phase_error: No help available

7.4.3.1.4.6 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.carrier.trace.perror.peak.clone()
```

Subgroups

7.4.3.1.4.7 StandardDev

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:PERRor:PEAK:SDEVIation
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:PERRor:PEAK:SDEVIation
```

class StandardDev

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:PERRor:PEAK:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.perror.peak.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak phase error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return phase_error: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:PERRor:PEAK:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.perror.peak.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak phase error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return phase_error: No help available

7.4.3.1.4.8 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:PERRor:PEAK:MAXimum
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:PERRor:PEAK:MAXimum
```

class Maximum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:PERRor:PEAK:MAXimum
value: List[float] = driver.multiEval.carrier.trace.perror.peak.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak phase error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return phase_error: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:PERRor:PEAK:MAXimum
value: List[float] = driver.multiEval.carrier.trace.perror.peak.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak phase error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement

length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return phase_error: No help available

7.4.3.1.4.9 Average

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:PELOR:PEAK:AVERAge
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:PELOR:PEAK:AVERAge
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:PELOR:PEAK:AVERAge
value: List[float] = driver.multiEval.carrier.trace.perror.peak.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak phase error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return phase_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:PELOR:PEAK:AVERAge
value: List[float] = driver.multiEval.carrier.trace.perror.peak.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak phase error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return phase_error: No help available

7.4.3.1.4.10 Current

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:PEERROR:PEAK:CURRENT
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:PEERROR:PEAK:CURRENT
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:PEERROR:PEAK:CURRENT
value: List[float] = driver.multiEval.carrier.trace.perror.peak.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the peak phase error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return phase_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:PEERROR:PEAK:CURRENT
value: List[float] = driver.multiEval.carrier.trace.perror.peak.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the peak phase error traces for up to 120 slots. Each current value is determined for a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return phase_error: No help available

7.4.3.1.5 CdPower

class CdPower

CdPower commands group definition. 50 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.3.1.5.1 Dpcch

class Dpcch

Dpcch 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.carrier.trace.cdPower.dpcch.clone()
```

Subgroups

7.4.3.1.5.2 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDPower:DPCCh:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:DPCCh:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPCCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:DPCCh:CURRENT
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.5.3 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACE:CDPower:DPCCh:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACE:CDPower:DPCCh:AVERage
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:DPCCh:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:DPCCh:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.5.4 Minimum

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACE:CDPower:DPCCh:MINimum
FETCH:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACE:CDPower:DPCCh:MINimum
```

class Minimum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:DPCCh:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.minimum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:DPCCh:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.minimum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.5.5 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACE:CDPower:DPCCh:MAXimum
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACE:CDPower:DPCCh:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:DPCCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPCCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.5.6 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:DPCCh:SDEviation
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:DPCCh:SDEviation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPCCh:SDEviation
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:CDPower:DPCCCh:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpcch.standardDev.
↪read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.5.7 Dpdch

class Dpdch

Dpdch 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.carrier.trace.cdPower.dpdch.clone()
```

Subgroups

7.4.3.1.5.8 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDPower:DPDCh:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↪:TRACe:CDPower:DPDCh:CURRent
```

class Current

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:DPDCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:DPDCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.5.9 Average**SCPI Commands**

```
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACE:CDPower:DPDCh:AVERage
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACE:CDPower:DPDCh:AVERage
```

class Average

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:CDPower:DPDCh:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.average.
↪fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:CDPower:DPDCh:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.average.
↪read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.5.10 Minimum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDPower:DPDCh:MINimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↪:TRACe:CDPower:DPDCh:MINimum
```

class Minimum

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.minimum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.minimum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.5.11 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDPower:DPDCh:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDPower:DPDCh:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.5.12 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDPower:DPDCh:SDEVIation
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDPower:DPDCh:SDEVIation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdPower.dpdch.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.5.13 Hsdpcch

class Hsdpcch

Hsdpcch 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.carrier.trace.cdPower.hsdpcch.clone()
```

Subgroups

7.4.3.1.5.14 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:HSDPcch:CURRENT
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:HSDPcch:CURRENT
```

class Current

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:CURRENT
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:CURRENT
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.5.15 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:HSDPcch:AVERAge
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:HSDPcch:AVERAge
```

class Average

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:AVERAge
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:AVERAge
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval. msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.5.16 Minimum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACE:CDPower:HSDPcch:MINimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACE:CDPower:HSDPcch:MINimum
```

class Minimum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:HSDPcch:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.minimum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval. msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:HSDPcch:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.minimum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.5.17 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:HSDPcch:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:HSDPcch:MAXimum
```

class Maximum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval. msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.5.18 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:HSDPcch:SDEVIation
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:HSDPcch:SDEVIation
```

class StandardDev

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval. msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdPower.hsdpcch.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.5.19 Edpcch

class Edpcch

Edpcch 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.carrier.trace.cdPower.edpcch.clone()
```

Subgroups

7.4.3.1.5.20 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:EDPCch:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDPower:EDPCch:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:CDPower:EDPCch:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.current.
↪read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.5.21 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↪:TRACe:CDPower:EDPCch:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↪:TRACe:CDPower:EDPCch:AVERage
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:CDPower:EDPCch:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.average.
↪fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.5.22 Minimum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDPower:EDPCch:MINimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDPower:EDPCch:MINimum
```

class Minimum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.minimum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.minimum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval. msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.5.23 Maximum

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDPower:EDPCch:MAXimum
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDPower:EDPCch:MAXimum
```

class Maximum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval. msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.5.24 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDPower:EDPCch:SDEVIation
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDPower:EDPCch:SDEVIation
```

class StandardDev

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpcch.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDP vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.5.25 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.multiEval.carrier.trace.cdPower.edpdch.repcap_edpdChannel_get()
driver.multiEval.carrier.trace.cdPower.edpdch.repcap_edpdChannel_set(repcap.EdpdChannel.
↳Nr1)
```

class Edpdch

Edpdch commands group definition. 10 total commands, 5 Sub-groups, 0 group commands Repeated Capability: EdpdChannel, default value after init: EdpdChannel.Nr1

Cloning the Group

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

Subgroups

7.4.3.1.5.26 Current

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↳<EdpdChannel>:CURRent
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↳<EdpdChannel>:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.current.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: No help available

read(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.current.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

7.4.3.1.5.27 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↳<EdpdChannel>:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↳<EdpdChannel>:AVERage
```

class Average

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

fetch(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.average.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

read(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.average.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

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

return edpdch: No help available

7.4.3.1.5.28 Minimum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↪<EdpdChannel>:MINimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↪<EdpdChannel>:MINimum
```

class Minimum

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

fetch(*carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:CDPower:EDPDch<nr>:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.minimum.
↪fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↪Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

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

return edpdch: No help available

read(*carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:MINimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.minimum.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: No help available

7.4.3.1.5.29 Maximum

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↳<EdpdChannel>:MAXimum
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↳<EdpdChannel>:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.maximum.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

read(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.maximum.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

7.4.3.1.5.30 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↳<EdpdChannel>:SDEviation
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDPower:EDPDch
↳<EdpdChannel>:SDEviation
```

class StandardDev

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

fetch(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:SDEviation
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.standardDev.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: No help available

read(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:CDPOWER:EDPDCH<nr>:SDEViation
value: List[float] = driver.multiEval.carrier.trace.cdPower.edpdch.standardDev.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDP vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: No help available

7.4.3.1.6 CdError

class CdError

CdError commands group definition. 40 total commands, 5 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.3.1.6.1 Dpcch

class Dpcch

Dpcch 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.carrier.trace.cdError.dpcch.clone()
```

Subgroups

7.4.3.1.6.2 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDERror:DPCCh:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDERror:DPCCh:CURRent
```

class Current

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:DPCCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdError.dpcch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:DPCCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdError.dpcch.current.
↳read(carrier = repcap.Carrier.Default)
```

(continues on next page)

(continued from previous page)

Returns the values of the RMS CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.6.3 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDERror:DPCCCh:AVERAge
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDERror:DPCCCh:AVERAge
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:DPCCCh:AVERAge
value: List[float] = driver.multiEval.carrier.trace.cdError.dpcch.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:DPCCCh:AVERAge
value: List[float] = driver.multiEval.carrier.trace.cdError.dpcch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod` modulation). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.6.4 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDERror:DPCCh:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDERror:DPCCh:MAXimum
```

class Maximum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:DPCCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.dpcch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod` modulation). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return dpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:DPCCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.dpcch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`, modulation). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.6.5 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACE:CDError:DPCh:SDEviation
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACE:CDError:DPCh:SDEviation
```

class StandardDev

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACE:CDError:DPCh:SDEviation
value: List[float] = driver.multiEval.carrier.trace.cdError.dpcch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`, modulation). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return dpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACE:CDError:DPCh:SDEviation
value: List[float] = driver.multiEval.carrier.trace.cdError.dpcch.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`. modulation) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.6.6 Dpdch

class Dpdch

Dpdch 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.carrier.trace.cdError.dpdch.clone()
```

Subgroups

7.4.3.1.6.7 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDError:DPDCh:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:DPDCh:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:DPDCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdError.dpdch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`. modulation) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return dpdch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↪:TRACE:CDError:DPDCh:CURRENT
value: List[float] = driver.multiEval.carrier.trace.cdError.dpdch.current.
↪read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return dpdch: No help available

7.4.3.1.6.8 Average

SCPI Commands

```
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:TRACE:CDError:DPDCh:AVERAge
FETCh:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>
↪:TRACE:CDError:DPDCh:AVERAge
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↪:TRACE:CDError:DPDCh:AVERAge
value: List[float] = driver.multiEval.carrier.trace.cdError.dpdch.average.
↪fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return dpdch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:CDError:DPDCh:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdError.dpdch.average.
↪read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.6.9 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDError:DPDCh:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↪:TRACe:CDError:DPDCh:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:CDError:DPDCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.dpdch.maximum.
↪fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDError:DPDCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.dpdch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.6.10 StandardDev

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACE:CDError:DPDCh:SDEviation
FETCh:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACE:CDError:DPDCh:SDEviation
```

class StandardDev

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDError:DPDCh:SDEviation
value: List[float] = driver.multiEval.carrier.trace.cdError.dpdch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(*carrier* =<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:DPDCh:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdError.dpdch.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.6.11 Hsdpcch

class Hsdpcch

Hsdpcch 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.carrier.trace.cdError.hsdpcch.clone()
```

Subgroups

7.4.3.1.6.12 Current

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:HSDPcch:CURRENT
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:HSDPcch:CURRENT
```

class Current

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

fetch(*carrier* =<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:CURRENT
value: List[float] = driver.multiEval.carrier.trace.cdError.hsdpcch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

(continues on next page)

(continued from previous page)

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdError.hsdpcch.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.6.13 Average

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDError:HSDPcch:AVERage
FETCh:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDError:HSDPcch:AVERage
```

class Average

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdError.hsdpcch.average.
↳fetch(carrier = repcap.Carrier.Default)
```

(continues on next page)

(continued from previous page)

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdError.hsdpcch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.6.14 Maximum

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDError:HSDPcch:MAXimum
FETCh:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDError:HSDPcch:MAXimum
```

class Maximum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.hsdpcch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

(continues on next page)

(continued from previous page)

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return hsdpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.hsdpcch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.6.15 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDError:HSDPcch:SDEViation
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:CDError:HSDPcch:SDEViation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:SDEViation
value: List[float] = driver.multiEval.carrier.trace.cdError.hsdpcch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

(continues on next page)

(continued from previous page)

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRAcE:CDError:HSDPcch:SDEviation
value: List[float] = driver.multiEval.carrier.trace.cdError.hsdpcch.standardDev.
↪read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.6.16 Edpcch

class Edpcch

Edpcch 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.carrier.trace.cdError.edpcch.clone()
```

Subgroups

7.4.3.1.6.17 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:EDPCch:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:EDPCch:CURRent
```

class Current

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdError.edpcch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return edpcch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdError.edpcch.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return edpcch: No help available

7.4.3.1.6.18 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:EDPCch:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:EDPCch:AVERage
```

class Average

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdError.edpcch.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`, `modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return edpcch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdError.edpcch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`, `modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return edpcch: No help available

7.4.3.1.6.19 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:EDPCch:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:EDPCch:MAXimum
```

class Maximum

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.edpcch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`, `modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return edpcch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.edpcch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`, `modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return edpcch: No help available

7.4.3.1.6.20 StandardDev

SCPI Commands

```

READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:EDPCch:SDEVIation
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:CDError:EDPCch:SDEVIation

```

class StandardDev

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```

# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdError.edpcch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)

```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return edpcch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```

# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.cdError.edpcch.standardDev.
↳read(carrier = repcap.Carrier.Default)

```

Returns the values of the RMS CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return edpcch: No help available

7.4.3.1.6.21 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.multiEval.carrier.trace.cdError.edpdch.repcap_edpdChannel_get()
driver.multiEval.carrier.trace.cdError.edpdch.repcap_edpdChannel_set(repcap.EdpdChannel.
↳Nr1)
```

class Edpdch

Edpdch commands group definition. 8 total commands, 4 Sub-groups, 0 group commands Repeated Capability: EdpdChannel, default value after init: EdpdChannel.Nr1

Cloning the Group

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

Subgroups

7.4.3.1.6.22 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDERror:EDPDch
↳<EdpdChannel>:CURRent
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDERror:EDPDch
↳<EdpdChannel>:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDERror:EDPDch<nr>:CURRent
value: List[float] = driver.multiEval.carrier.trace.cdError.edpdch.current.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float RMS CDE trace results, one result per measured slot or half-slot
Range: -100 dB to 0 dB (SDEViation: 0 dB to 50 dB) , Unit: dB

read(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPDch<nr>:CURRENT
value: List[float] = driver.multiEval.carrier.trace.cdError.edpdch.current.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Modulation, CDP and CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float RMS CDE trace results, one result per measured slot or half-slot
Range: -100 dB to 0 dB (SDEViation: 0 dB to 50 dB) , Unit: dB

7.4.3.1.6.23 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDError:EDPDch
↳<EdpdChannel>:AVERage
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:CDError:EDPDch
↳<EdpdChannel>:AVERage
```

class Average

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

fetch(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPDch<nr>:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdError.edpdch.average.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement

length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: float RMS CDE trace results, one result per measured slot or half-slot
Range: -100 dB to 0 dB (SDEVIation: 0 dB to 50 dB) , Unit: dB

read(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:EDPDch<nr>:AVERage
value: List[float] = driver.multiEval.carrier.trace.cdError.edpdch.average.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: float RMS CDE trace results, one result per measured slot or half-slot
Range: -100 dB to 0 dB (SDEVIation: 0 dB to 50 dB) , Unit: dB

7.4.3.1.6.24 Maximum

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDError:EDPDch
↳<EdpdChannel>:MAXimum
FETCH:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDError:EDPDch
↳<EdpdChannel>:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPDch<nr>:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.edpdch.maximum.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: float RMS CDE trace results, one result per measured slot or half-slot
Range: -100 dB to 0 dB (SDEVIation: 0 dB to 50 dB), Unit: dB

read(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPDch<nr>:MAXimum
value: List[float] = driver.multiEval.carrier.trace.cdError.edpdch.maximum.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: float RMS CDE trace results, one result per measured slot or half-slot
Range: -100 dB to 0 dB (SDEVIation: 0 dB to 50 dB), Unit: dB

7.4.3.1.6.25 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDERror:EDPDch
↳<EdpdChannel>:SDEViation
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:CDERror:EDPDch
↳<EdpdChannel>:SDEViation
```

class StandardDev

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

fetch(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:EDPDch<nr>:SDEViation
value: List[float] = driver.multiEval.carrier.trace.cdError.edpdch.standardDev.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: float RMS CDE trace results, one result per measured slot or half-slot
Range: -100 dB to 0 dB (SDEViation: 0 dB to 50 dB), Unit: dB

read(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:EDPDch<nr>:SDEViation
value: List[float] = driver.multiEval.carrier.trace.cdError.edpdch.standardDev.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the RMS CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: float RMS CDE trace results, one result per measured slot or half-slot
Range: -100 dB to 0 dB (SDEViation: 0 dB to 50 dB) , Unit: dB

7.4.3.1.7 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.carrier.trace.freqError.clone()
```

Subgroups

7.4.3.1.7.1 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:FERRor:SDEViation
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:FERRor:SDEViation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:FERRor:SDEViation
value: List[float] = driver.multiEval.carrier.trace.freqError.standardDev.
↪fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the carrier frequency error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return frequency_error: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:FERRor:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.freqError.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the carrier frequency error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return frequency_error: No help available

7.4.3.1.7.2 Maximum

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:FERRor:MAXimum
FETCh:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:FERRor:MAXimum
```

class Maximum

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:FERRor:MAXimum
value: List[float] = driver.multiEval.carrier.trace.freqError.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the carrier frequency error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return frequency_error: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:FERRor:MAXimum
value: List[float] = driver.multiEval.carrier.trace.freqError.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the carrier frequency error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return frequency_error: No help available

7.4.3.1.7.3 Average

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:FERRor:AVERage
FETCh:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:FERRor:AVERage
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:FERRor:AVERage
value: List[float] = driver.multiEval.carrier.trace.freqError.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the carrier frequency error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return frequency_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:FERRor:AVERage
value: List[float] = driver.multiEval.carrier.trace.freqError.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the carrier frequency error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return frequency_error: No help available

7.4.3.1.7.4 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:FERRor:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:FERRor:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:FERRor:CURRent
value: List[float] = driver.multiEval.carrier.trace.freqError.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the carrier frequency error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return frequency_error: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:FERRor:CURRent
value: List[float] = driver.multiEval.carrier.trace.freqError.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the carrier frequency error traces for up to 120 slots. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Modulation, CDP and CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return frequency_error: No help available

7.4.3.1.8 Psteps

class Psteps

Psteps 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.carrier.trace.psteps.clone()
```

Subgroups

7.4.3.1.8.1 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:PSTeps:SDEViation
FETCH:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:PSTeps:SDEViation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:PSTeps:SDEViation
value: List[float] = driver.multiEval.carrier.trace.psteps.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation)

. As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return power_steps: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRAcE:PSTeps:SDEViation
value: List[float] = driver.multiEval.carrier.trace.psteps.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return power_steps: No help available

7.4.3.1.8.2 Minimum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRAcE:PSTeps:MINimum
FETCH:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRAcE:PSTeps:MINimum
```

class Minimum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRAcE:PSTeps:MINimum
value: List[float] = driver.multiEval.carrier.trace.psteps.minimum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending

on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return power_steps: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:PSTeps:MINimum
value: List[float] = driver.multiEval.carrier.trace.psteps.minimum.read(carrier_
↳= repcap.Carrier.Default)
```

Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return power_steps: No help available

7.4.3.1.8.3 Maximum

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:PSTeps:MAXimum
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:PSTeps:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:PSTeps:MAXimum
value: List[float] = driver.multiEval.carrier.trace.psteps.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return power_steps: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRAcE:PSTeps:MAXimum
value: List[float] = driver.multiEval.carrier.trace.psteps.maximum.read(carrier_
↳= repcap.Carrier.Default)
```

Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return power_steps: No help available

7.4.3.1.8.4 Average

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRAcE:PSTeps:AVERage
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRAcE:PSTeps:AVERage
```

class Average

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRAcE:PSTeps:AVERage
value: List[float] = driver.multiEval.carrier.trace.psteps.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return power_steps: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳ :TRAcE:PSTeps:AVERage
value: List[float] = driver.multiEval.carrier.trace.psteps.average.read(carrier_
↳ = repcap.Carrier.Default)
```

Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return power_steps: No help available

7.4.3.1.8.5 Current

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRAcE:PSTeps:CURRent
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRAcE:PSTeps:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳ :TRAcE:PSTeps:CURRent
value: List[float] = driver.multiEval.carrier.trace.psteps.current.
↳ fetch(carrier = repcap.Carrier.Default)
```


Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return power_steps: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:PSTEPS:CURRENT
value: List[float] = driver.multiEval.carrier.trace.psteps.current.read(carrier_
↳ = repcap.Carrier.Default)
```

Returns the values of the UE power step traces for up to 120 slots. Each power step is calculated as the difference between the UE power of a half-slot or full-slot and the preceding half-slot or full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . As there is no previous slot / halfslot for slot 0, the first returned power step value equals NCAP. The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) . The results of the current, average, minimum, maximum and standard deviation traces can be retrieved. The minimum and standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return power_steps: No help available

7.4.3.1.9 RcdError

class RcdError

RcdError commands group definition. 50 total commands, 6 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.3.1.9.1 Sf

class Sf

Sf 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.carrier.trace.rcdError.sf.clone()
```

Subgroups

7.4.3.1.9.2 Dpcch

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:SF:DPCCh
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:SF:DPCCh
```

class Dpcch

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

fetch(carrier=<Carrier.Default: -1>) → List[int]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:SF:DPCCh
value: List[int] = driver.multiEval.carrier.trace.rcdError.sf.dpcch.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the current spreading factors for the DPCCH and the DPDCH. Each value refers to a half-slot or a full-slot, depending on the measurement period (method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[int]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:SF:DPCCh
value: List[int] = driver.multiEval.carrier.trace.rcdError.sf.dpcch.
↳read(carrier = repcap.Carrier.Default)
```

Returns the current spreading factors for the DPCCH and the DPDCH. Each value refers to a half-slot or a full-slot, depending on the measurement period (method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return dpcch: No help available

7.4.3.1.9.3 Dpdch

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:SF:DPDCh
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:SF:DPDCh
```

class Dpdch

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

fetch(carrier=<Carrier.Default: -1>) → List[int]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:RCDerror:SF:DPDCh
value: List[int] = driver.multiEval.carrier.trace.rcdError.sf.dpdch.
↪fetch(carrier = repcap.Carrier.Default)
```

Returns the current spreading factors for the DPCCH and the DPDCH. Each value refers to a half-slot or a full-slot, depending on the measurement period (method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return dpdch: No help available

read(carrier=<Carrier.Default: -1>) → List[int]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:RCDerror:SF:DPDCh
value: List[int] = driver.multiEval.carrier.trace.rcdError.sf.dpdch.
↪read(carrier = repcap.Carrier.Default)
```

Returns the current spreading factors for the DPCCH and the DPDCH. Each value refers to a half-slot or a full-slot, depending on the measurement period (method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return dpdch: No help available

7.4.3.1.9.4 Hsdpcch

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:SF:HSDPcch
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:SF:HSDPcch
```

class Hsdpcch

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

fetch(*carrier=<Carrier.Default: -1>*) → List[int]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:RCDerror:SF:HSDPcch
value: List[int] = driver.multiEval.carrier.trace.rcdError.sf.hsdpcch.
↪fetch(carrier = repcap.Carrier.Default)
```

Returns the current spreading factors for the E-DPCCH and the HS-DPCCH. Each value refers to a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[int]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:RCDerror:SF:HSDPcch
value: List[int] = driver.multiEval.carrier.trace.rcdError.sf.hsdpcch.
↪read(carrier = repcap.Carrier.Default)
```

Returns the current spreading factors for the E-DPCCH and the HS-DPCCH. Each value refers to a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return hsdpcch: No help available

7.4.3.1.9.5 Edpcch

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:SF:EDPCch
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:SF:EDPCch
```

class Edpcch

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

fetch(*carrier*=<Carrier.Default: -1>) → List[int]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:SF:EDPCch
value: List[int] = driver.multiEval.carrier.trace.rcdError.sf.edpcch.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the current spreading factors for the E-DPCCH and the HS-DPCCH. Each value refers to a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[int]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:SF:EDPCch
value: List[int] = driver.multiEval.carrier.trace.rcdError.sf.edpcch.
↳read(carrier = repcap.Carrier.Default)
```

Returns the current spreading factors for the E-DPCCH and the HS-DPCCH. Each value refers to a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.9.6 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.multiEval.carrier.trace.rcdError.sf.edpdch.repcap_edpdChannel_get()
driver.multiEval.carrier.trace.rcdError.sf.edpdch.repcap_edpdChannel_set(repcap.
↳ EdpdChannel.Nr1)
```

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:RCDerror:SF:EDPDch
↳ <EdpdChannel>
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:TRACE:RCDerror:SF:EDPDch
↳ <EdpdChannel>
```

class Edpdch

Edpdch commands group definition. 2 total commands, 0 Sub-groups, 2 group commands Repeated Capability: EdpdChannel, default value after init: EdpdChannel.Nr1

fetch(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳ :TRACE:RCDerror:SF:EDPDch<nr>
value: List[float] = driver.multiEval.carrier.trace.rcdError.sf.edpdch.
↳ fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳ Default)
```

Returns the spreading factors for the E-DPDCH 1 to 4. Each current value refers to a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount).

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

read(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳ :TRACE:RCDerror:SF:EDPDch<nr>
value: List[float] = driver.multiEval.carrier.trace.rcdError.sf.edpdch.
↳ read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳ Default)
```

Returns the spreading factors for the E-DPDCH 1 to 4. Each current value refers to a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcd-

maMeas.Configure.MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount) .

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.carrier.trace.rcdError.sf.edpdch.clone()
```

7.4.3.1.9.7 Dpcch

class Dpcch

Dpcch 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.carrier.trace.rcdError.dpcch.clone()
```

Subgroups

7.4.3.1.9.8 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:RCDerror:DPCCh:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:RCDerror:DPCCh:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpcch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpcch.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.9.9 Average

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPCCh:AVERage
FETCh:WCDMA:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPCCh:AVERage
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:AVERage
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpcch.average.
↳fetch(carrier = repcap.Carrier.Default)
```


Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:AVERAge
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpcch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.9.10 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPCCh:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPCCh:MAXimum
```

class Maximum

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpcch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`. modulation) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return dpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpcch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`. modulation) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.9.11 StandardDev

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPCCh:SDEViation
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPCCh:SDEViation
```

class StandardDev

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:SDEViation
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpcch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`. modulation) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return dpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:RCDError:DPCCh:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpcch.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod`. modulation) . The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return dpcch: No help available

7.4.3.1.9.12 Dpdch

class Dpdch

Dpdch 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.carrier.trace.rcdError.dpdch.clone()
```

Subgroups

7.4.3.1.9.13 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPDCh:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPDCh:CURRent
```

class Current

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpdch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:CURRent
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpdch.current.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.9.14 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPDCh:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPDCh:AVERage
```

class Average

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:AVERage
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpdch.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:AVERage
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpdch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.9.15 Maximum

SCPI Commands

```

READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPDCh:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPDCh:MAXimum

```

class Maximum

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```

# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpdch.maximum.
↳fetch(carrier = repcap.Carrier.Default)

```

Returns the values of the relative CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```

# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpdch.maximum.
↳read(carrier = repcap.Carrier.Default)

```

Returns the values of the relative CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.9.16 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPDCh:SDEVIation
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:DPDCh:SDEVIation
```

class StandardDev

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpdch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.rcdError.dpdch.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the DPCCCH and the DPDCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return dpdch: No help available

7.4.3.1.9.17 Hsdpcch

class Hsdpcch

Hsdpcch 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.carrier.trace.rcdError.hsdpcch.clone()
```

Subgroups

7.4.3.1.9.18 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:RCDerror:HSDPcch:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↳:TRACe:RCDerror:HSDPcch:CURRent
```

class Current

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

fetch(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:HSDPcch:CURRent
value: List[float] = driver.multiEval.carrier.trace.rcdError.hsdpcch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure. MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier*=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:HSDPcch:CURRent
value: List[float] = driver.multiEval.carrier.trace.rcdError.hsdpcch.current.
↳read(carrier = repcap.Carrier.Default)
```


Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.9.19 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACE:RCDerror:HSDPcch:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACE:RCDerror:HSDPcch:AVERage
```

class Average

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACE:RCDerror:HSDPcch:AVERage
value: List[float] = driver.multiEval.carrier.trace.rcdError.hsdpcch.average.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACE:RCDerror:HSDPcch:AVERage
value: List[float] = driver.multiEval.carrier.trace.rcdError.hsdpcch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.9.20 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACE:RCDerror:HSDPcch:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACE:RCDerror:HSDPcch:MAXimum
```

class Maximum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACE:RCDerror:HSDPcch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.hsdpcch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACE:RCDerror:HSDPcch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.hsdpcch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.9.21 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDError:HSDPcch:SDEViation
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDError:HSDPcch:SDEViation
```

class StandardDev

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDError:HSDPcch:SDEViation
value: List[float] = driver.multiEval.carrier.trace.rcdError.hsdpcch.
↳standardDev.fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDError:HSDPcch:SDEViation
value: List[float] = driver.multiEval.carrier.trace.rcdError.hsdpcch.
↳standardDev.read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

return hsdpcch: No help available

7.4.3.1.9.22 Edpcch

class Edpcch

Edpcch 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.carrier.trace.rcdError.edpcch.clone()
```

Subgroups

7.4.3.1.9.23 Current

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
↳:TRACE:RCDError:EDPCch:CURRENT
FETCh:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
↳:TRACE:RCDError:EDPCch:CURRENT
```

class Current

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDError:EDPCch:CURRENT
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpcch.current.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:RCDerror:EDPCch:CURRent
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpcch.current.
↪read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure. MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Relative CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.9.24 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↪:TRACe:RCDerror:EDPCch:AVERAge
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>
↪:TRACe:RCDerror:EDPCch:AVERAge
```

class Average

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

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:RCDerror:EDPCch:AVERAge
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpcch.average.
↪fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure. MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Relative CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPCch:AVERage
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpcch.average.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure. MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Relative CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.9.25 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:EDPCch:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:EDPCch:MAXimum
```

class Maximum

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPCch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpcch.maximum.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure. MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Relative CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPCch:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpcch.maximum.
↳read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure. MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Relative CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.9.26 StandardDev

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:EDPCch:SDEviation
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>
↳:TRACe:RCDerror:EDPCch:SDEviation
```

class StandardDev

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

fetch(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPCch:SDEviation
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpcch.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure. MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Relative CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

read(*carrier=<Carrier.Default: -1>*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACE:RCDError:EDPCch:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpcch.standardDev.
↪read(carrier = repcap.Carrier.Default)
```

Returns the values of the relative CDE vs. slot traces for the HS-DPCCH and the E-DPCCH. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure. MultiEval.msCount) . The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation traces cannot be displayed at the GUI. See also 'Detailed Views: Relative CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return edpcch: No help available

7.4.3.1.9.27 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.multiEval.carrier.trace.rcdError.edpdch.repcap_edpdChannel_get()
driver.multiEval.carrier.trace.rcdError.edpdch.repcap_edpdChannel_set(repcap.EdpdChannel.
↪Nr1)
```

class Edpdch

Edpdch commands group definition. 8 total commands, 4 Sub-groups, 0 group commands Repeated Capability: EdpdChannel, default value after init: EdpdChannel.Nr1

Cloning the Group

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


Subgroups

7.4.3.1.9.28 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:EDPDch
↳<EdpdChannel>:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:EDPDch
↳<EdpdChannel>:CURRent
```

class Current

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

fetch(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:CURRent
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpdch.current.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the relative CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: No help available

read(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:CURRent
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpdch.current.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the relative CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

7.4.3.1.9.29 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:RCDerror:EDPDch
↳<EdpdChannel>:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:RCDerror:EDPDch
↳<EdpdChannel>:AVERage
```

class Average

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

fetch(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:AVERage
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpdch.average.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the relative CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Relative CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

read(*carrier*=<Carrier.Default: -1>, *edpdChannel*=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:AVERage
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpdch.average.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the relative CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

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

return edpdch: No help available

7.4.3.1.9.30 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:RCDerror:EDPDch
↳<EdpdChannel>:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:TRACe:RCDerror:EDPDch
↳<EdpdChannel>:MAXimum
```

class Maximum

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

fetch(*carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpdch.maximum.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the relative CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method `RsCmwWcdmaMeas.Configure.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCmwWcdmaMeas.Configure.MultiEval.msCount`). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

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

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

return edpdch: No help available

read(*carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:MAXimum
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpdch.maximum.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the relative CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

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

return edpdch: No help available

7.4.3.1.9.31 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:EDPDch
↳<EdpdChannel>:SDEVIation
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:TRACe:RCDerror:EDPDch
↳<EdpdChannel>:SDEVIation
```

class StandardDev

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

fetch(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpdch.standardDev.
↳fetch(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the relative CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval. msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also ‘Detailed Views: Relative CDE’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

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

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

read(carrier=<Carrier.Default: -1>, edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDError:EDPDch<nr>:SDEVIation
value: List[float] = driver.multiEval.carrier.trace.rcdError.edpdch.standardDev.
↳read(carrier = repcap.Carrier.Default, edpdChannel = repcap.EdpdChannel.
↳Default)
```

Returns the values of the relative CDE vs. slot traces for the E-DPDCH 1 to 4. Each current value is averaged over a half-slot or a full-slot, depending on the measurement period (see method RsCmwWcdmaMeas.Configure.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCmwWcdmaMeas.Configure.MultiEval.msCount). The results of the current, average, maximum and standard deviation traces can be retrieved. The standard deviation trace cannot be displayed at the GUI. See also 'Detailed Views: Relative CDE'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: No help available

7.4.3.2 Modulation

class Modulation

Modulation commands group definition. 12 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.3.2.1 StandardDev

SCPI Commands

```
CALCulate:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:MODulation:SDEVIation
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:MODulation:SDEVIation
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>: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 and peak value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value Range: 0 % to 100 %, Unit: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float User equipment power step Range: -50 dB to 50 dB, Unit: dB
- Phase_Disc: enums.ResultStatus2: float Phase discontinuity Range: -180 deg to 180 deg, Unit: deg

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- 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: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float User equipment power step Range: -50 dB to 50 dB, Unit: dB
- Phase_Disc: float: float Phase discontinuity Range: -180 deg to 180 deg, Unit: deg
- Tx_Time_Alignment: float: No parameter help available

calculate(*carrier*=<*Carrier.Default*: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:SDEVIation
value: CalculateStruct = driver.multiEval.carrier.modulation.standardDev.
↳calculate(carrier = repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCh and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

fetch(*carrier=<Carrier.Default: -1>*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:SDEVIation
value: ResultData = driver.multiEval.carrier.modulation.standardDev.
↳fetch(carrier = repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCh and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier=<Carrier.Default: -1>*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:SDEVIation
value: ResultData = driver.multiEval.carrier.modulation.standardDev.
↳read(carrier = repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCh and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.2.2 Maximum

SCPI Commands

```
CALCulate:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:MODulation:MAXimum
READ:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>:MODulation:MAXimum
FETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Carrier>: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 and peak value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value Range: 0 % to 100 %, Unit: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float User equipment power step Range: -50 dB to 50 dB, Unit: dB
- Phase_Disc: enums.ResultStatus2: float Phase discontinuity Range: -180 deg to 180 deg, Unit: deg

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- 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: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available

- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float User equipment power step Range: -50 dB to 50 dB, Unit: dB
- Phase_Disc: float: float Phase discontinuity Range: -180 deg to 180 deg, Unit: deg
- Tx_Time_Alignment: float: No parameter help available

calculate(carrier=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:MAXimum
value: CalculateStruct = driver.multiEval.carrier.modulation.maximum.
↳calculate(carrier = repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCH and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

fetch(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:MAXimum
value: ResultData = driver.multiEval.carrier.modulation.maximum.fetch(carrier =
↳repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCH and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:MAXimum
value: ResultData = driver.multiEval.carrier.modulation.maximum.read(carrier =
↳repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCH and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.2.3 Current

SCPI Commands

CALCulate:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:MODulation:CURRent
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:MODulation:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>: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 and peak value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value Range: 0 % to 100 %, Unit: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float User equipment power step Range: -50 dB to 50 dB, Unit: dB
- Phase_Disc: float: float Phase discontinuity Range: -180 deg to 180 deg, Unit: deg

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- 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: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float User equipment power step Range: -50 dB to 50 dB, Unit: dB
- Phase_Disc: float: float Phase discontinuity Range: -180 deg to 180 deg, Unit: deg
- Tx_Time_Alignment: float: No parameter help available

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:CURRent
value: CalculateStruct = driver.multiEval.carrier.modulation.current.
↳calculate(carrier = repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCh and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:CURRent
value: ResultData = driver.multiEval.carrier.modulation.current.fetch(carrier =
↳repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCh and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:CURRent
value: ResultData = driver.multiEval.carrier.modulation.current.read(carrier =
↳repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check

results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCH and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.2.4 Average

SCPI Commands

CALCulate:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:MODulation:AVERage
 FETCH:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:MODulation:AVERage
 READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>: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 and peak value Range: 0 % to 100 %, Unit: %
- Evm_Peak: float: float Error vector magnitude RMS and peak value Range: 0 % to 100 %, Unit: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float User equipment power step Range: -50 dB to 50 dB, Unit: dB
- Phase_Disc: enums.ResultStatus2: float Phase discontinuity Range: -180 deg to 180 deg, Unit: deg

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- 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: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %

- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float User equipment power step Range: -50 dB to 50 dB, Unit: dB
- Phase_Disc: float: float Phase discontinuity Range: -180 deg to 180 deg, Unit: deg
- Tx_Time_Alignment: float: No parameter help available

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:AVERage
value: CalculateStruct = driver.multiEval.carrier.modulation.average.
↳calculate(carrier = repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCH and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:AVERage
value: ResultData = driver.multiEval.carrier.modulation.average.fetch(carrier =
↳repcap.Carrier.Default)
```

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCH and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:AVERage
value: ResultData = driver.multiEval.carrier.modulation.average.read(carrier =
↳repcap.Carrier.Default)
```

(continues on next page)

(continued from previous page)

Return the current, average, maximum and standard deviation single value results. The return values described below are returned by FETCH and READ commands. CALCulate commands return limit check results instead, one value for each of the first 14 results listed below. The TX time alignment is only returned by FETCH and READ commands.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.3 RcdError

class RcdError

RcdError commands group definition. 16 total commands, 6 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.3.3.1 Current

SCPI Commands

```
CALCulate:WCDMa:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:RCDerror:CURRENT
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:RCDerror:CURRENT
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:RCDerror: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'
- Dpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Edpdch_2: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:RCDError:CURRENT
value: CalculateStruct = driver.multiEval.carrier.rcdError.current.
↳calculate(carrier = repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:RCDError:CURRENT
value: ResultData = driver.multiEval.carrier.rcdError.current.fetch(carrier =
↳repcap.Carrier.Default) (continues on next page)
```

(continued from previous page)

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier=<Carrier.Default: -1>*) → ResultData

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:RCDError:CURRENT
value: ResultData = driver.multiEval.carrier.rcdError.current.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.3.2 Average

SCPI Commands

```
CALCulate:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:RCDError:AVERAge
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:RCDError:AVERAge
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:RCDError: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'
- Dpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Edpdch_1: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:RCDerror:AVERage
value: CalculateStruct = driver.multiEval.carrier.rcdError.average.
↪calculate(carrier = repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:RCDerror:AVERage
value: ResultData = driver.multiEval.carrier.rcdError.average.fetch(carrier =
↳repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:RCDerror:AVERage
value: ResultData = driver.multiEval.carrier.rcdError.average.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.3.3 Maximum

SCPI Commands

```
CALCulate:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:RCDerror:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:RCDerror:MAXimum
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:RCDerror: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'
- Dpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Edpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:RCDError:MAXimum
value: CalculateStruct = driver.multiEval.carrier.rcdError.maximum.
↪calculate(carrier = repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:RCERROR:MAXIMUM
value: ResultData = driver.multiEval.carrier.rcdError.maximum.fetch(carrier =
↳repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:RCERROR:MAXIMUM
value: ResultData = driver.multiEval.carrier.rcdError.maximum.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.3.4 StandardDev

SCPI Commands

```
CALCULATE:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:RCERROR:SDEVATION
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:RCERROR:SDEVATION
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:CARRIER<Carrier>:RCERROR:SDEVATION
```

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'
- Dpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Hsdpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RCDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:RCDError:SDEViation
value: CalculateStruct = driver.multiEval.carrier.rcdError.standardDev.
↪calculate(carrier = repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:RCDerror:SDEViation
value: ResultData = driver.multiEval.carrier.rcdError.standardDev.fetch(carrier,
↳= repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

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

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:RCDerror:SDEViation
value: ResultData = driver.multiEval.carrier.rcdError.standardDev.read(carrier,
↳= repcap.Carrier.Default)
```

Returns the RCDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values 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.

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

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

7.4.3.3.5 Sf

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:RCDerror:SF
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:RCDerror:SF
```

class Sf

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Dpcch: int: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 Spreading factors for the indicated channels
- Dpdch: int: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 Spreading factors for the indicated channels
- Hsdpcch: int: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 Spreading factors for the indicated channels

- Edpcch: int: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 Spreading factors for the indicated channels
- Edpdch_1: int: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 Spreading factors for the indicated channels
- Edpdch_2: int: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 Spreading factors for the indicated channels
- Edpdch_3: int: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 Spreading factors for the indicated channels
- Edpdch_4: int: 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 Spreading factors for the indicated channels

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:RCDerror:SF
value: ResultData = driver.multiEval.carrier.rcdError.sf.fetch(carrier = repcap.
↪Carrier.Default)
```

Returns the spreading factors of the dedicated physical channels determined from a selected slot. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

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

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:RCDerror:SF
value: ResultData = driver.multiEval.carrier.rcdError.sf.read(carrier = repcap.
↪Carrier.Default)
```

Returns the spreading factors of the dedicated physical channels determined from a selected slot. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

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

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

7.4.3.3.6 OcInfo

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:RCDerror:OCInfo
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:RCDerror:OCInfo
```

class OcInfo

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’

- State: List[enums.State]: OFF | VAR | ON State of the channel OFF: Channel off since start of measurement VAR: Channel has been on and off ON: Channel on since start of measurement
- Spreading_Factor: List[enums.SpreadingFactorB]: No parameter help available
- Modulation: List[enums.Modulation]: BPSK | 4PAM | 4PVar Modulation type of the channel BPSK: Constantly BPSK modulated 4PAM: Constantly 4PAM modulated 4PVar: BPSK and 4PAM occurred

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:RCDerror:OCInfo
value: ResultData = driver.multiEval.carrier.rcdError.ocInfo.fetch(carrier =
↳repcap.Carrier.Default)
```

Returns the overall channel information for the RCDE measurement. This information is determined from all measurements.

INTRO_CMD_HELP: The parameters <State>, <SpreadFactor> and <Modulation> are returned for the individual channels:

- Values 2 to 4: DPCCH
- Values 5 to 7: DPDCH
- Values 8 to 10: HSDPCCH
- Values 11 to 13: EDPCCCH
- Values 14 to 16: EDPDCH1
- Values 17 to 19: EDPDCH2
- Values 20 to 22: EDPDCH3
- Values 23 to 25: EDPDCH4

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:RCDerror:OCInfo
value: ResultData = driver.multiEval.carrier.rcdError.ocInfo.read(carrier =
↳repcap.Carrier.Default)
```

Returns the overall channel information for the RCDE measurement. This information is determined from all measurements.

INTRO_CMD_HELP: The parameters <State>, <SpreadFactor> and <Modulation> are returned for the individual channels:

- Values 2 to 4: DPCCH
- Values 5 to 7: DPDCH
- Values 8 to 10: HSDPCCH
- Values 11 to 13: EDPCCCH
- Values 14 to 16: EDPDCH1

- Values 17 to 19: EDPDCH2
- Values 20 to 22: EDPDCH3
- Values 23 to 25: EDPDCH4

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.4 CdPower

class CdPower

CdPower 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.carrier.cdPower.clone()
```

Subgroups

7.4.3.4.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDPower:CURRENT
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDPower: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'
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:CURRent
value: ResultData = driver.multiEval.carrier.cdPower.current.fetch(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:CURRent
value: ResultData = driver.multiEval.carrier.cdPower.current.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.4.2 Average

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDPower:AVERage
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDPower:AVERage
```

class Average

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB

- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:CDPower:AVERAGE
value: ResultData = driver.multiEval.carrier.cdPower.average.fetch(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:CDPower:AVERAGE
value: ResultData = driver.multiEval.carrier.cdPower.average.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.4.3 Minimum

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDPower:MINimum
FETCh:WCDma:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDPower:MINimum
```

class Minimum

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:CDPower:MINimum
value: ResultData = driver.multiEval.carrier.cdPower.minimum.fetch(carrier = ↪
↪repcap.Carrier.Default)
```

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:CDPower:MINimum
value: ResultData = driver.multiEval.carrier.cdPower.minimum.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.4.4 Maximum

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:MAXimum
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:MAXimum
```

class Maximum

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:CDPower:MAXimum
value: ResultData = driver.multiEval.carrier.cdPower.maximum.fetch(carrier =
↳repcap.Carrier.Default)
```

(continues on next page)

(continued from previous page)

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier=<Carrier.Default: -1>*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:MAXimum
value: ResultData = driver.multiEval.carrier.cdPower.maximum.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.4.5 StandardDev

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDPower:SDEViation
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDPower:SDEViation
```

class StandardDev

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:SDEViation
value: ResultData = driver.multiEval.carrier.cdPower.standardDev.fetch(carrier,
↳= repcap.Carrier.Default)
```

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:SDEViation
value: ResultData = driver.multiEval.carrier.cdPower.standardDev.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDP vs. slot values measured in a selected slot. In addition to the current values, average, minimum, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.5 CdError

class CdError

CdError 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.carrier.cdError.clone()
```

Subgroups

7.4.3.5.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:CDERror:CURRent
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:CDERror: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'
- Dpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:CDERror:CURRent
value: ResultData = driver.multiEval.carrier.cdError.current.fetch(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData


```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:CDERror:CURRent
value: ResultData = driver.multiEval.carrier.cdError.current.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.5.2 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:CDERror:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEValuation:CARRier<Carrier>:CDERror:AVERage
```

class Average

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:CDERror:AVERage
value: ResultData = driver.multiEval.carrier.cdError.average.fetch(carrier =
↳repcap.Carrier.Default)
```

(continues on next page)

(continued from previous page)

Returns the RMS CDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values can be retrieved.

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

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

read(*carrier=<Carrier.Default: -1>*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:CDERror:AVERage
value: ResultData = driver.multiEval.carrier.cdError.average.read(carrier = ↪
↪repcap.Carrier.Default)
```

Returns the RMS CDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values can be retrieved.

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

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

7.4.3.5.3 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDERror:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDERror:MAXimum
```

class Maximum

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Dpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Edpdch_3: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDERror:MAXimum
value: ResultData = driver.multiEval.carrier.cdError.maximum.fetch(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDERror:MAXimum
value: ResultData = driver.multiEval.carrier.cdError.maximum.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.3.5.4 StandardDev

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDERror:SDEVIation
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRier<Carrier>:CDERror:SDEVIation
```

class StandardDev

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Dpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB

- Hsdpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:CDERror:SDEViation
value: ResultData = driver.multiEval.carrier.cdError.standardDev.fetch(carrier,
↳repcap.Carrier.Default)
```

Returns the RMS CDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:CDERror:SDEViation
value: ResultData = driver.multiEval.carrier.cdError.standardDev.read(carrier =
↳repcap.Carrier.Default)
```

Returns the RMS CDE vs. slot values measured in a selected slot. In addition to the current values, average, maximum and standard deviation values can be retrieved.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

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

7.4.4 Spectrum

class Spectrum

Spectrum 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.spectrum.clone()
```

Subgroups

7.4.4.1 Average

SCPI Commands

```
CALCulate:WCDMa:MEASurement<Instance>:MEValuation:SPECTrum:AVERage
FETCh:WCDMa:MEASurement<Instance>:MEValuation:SPECTrum:AVERage
READ:WCDMa:MEASurement<Instance>:MEValuation:SPECTrum: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'
- Carrier_Power: float: float Power at the nominal carrier UL frequency Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Obw: float: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- Sem_Margin_Abij: float: No parameter help available
- Sem_Margin_Bcjk: float: No parameter help available
- Sem_Margin_Cdkl: float: No parameter help available
- Sem_Margin_Efmn: float: No parameter help available
- Sem_Margin_Fenm: float: No parameter help available

- Sem_Margin_Dclk: float: No parameter help available
- Sem_Margin_Cbkj: float: No parameter help available
- Sem_Margin_Baji: float: No parameter help available
- Ue_Power: enums.ResultStatus2: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Emask_Margin_Ad: float: No parameter help available
- Emask_Margin_Da: float: No parameter help available
- Carrier_Power_L: enums.ResultStatus2: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm
- Carrier_Power_R: enums.ResultStatus2: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Carrier_Power: float: float Power at the nominal carrier UL frequency Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Obw: float: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- Sem_Margin_Abij: float: No parameter help available
- Sem_Margin_Bcjk: float: No parameter help available
- Sem_Margin_Cdkl: float: No parameter help available
- Sem_Margin_Efmn: float: No parameter help available
- Sem_Margin_Fenm: float: No parameter help available
- Sem_Margin_Dclk: float: No parameter help available
- Sem_Margin_Cbkj: float: No parameter help available
- Sem_Margin_Baji: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Sem_Margin_Ad: float: No parameter help available
- Sem_Margin_Da: float: No parameter help available

- Sem_Abij_At_Freq: float: No parameter help available
- Sem_Bcjk_At_Freq: float: No parameter help available
- Sem_Cdkl_At_Freq: float: No parameter help available
- Sem_Efmn_At_Freq: float: No parameter help available
- Sem_Fenm_At_Freq: float: No parameter help available
- Sem_Dclk_At_Freq: float: No parameter help available
- Sem_Cbkj_At_Freq: float: No parameter help available
- Sem_Baji_At_Freq: float: No parameter help available
- Sem_Adat_Freq: float: No parameter help available
- Sem_Da_At_Freq: float: No parameter help available
- Carrier_Power_L: float: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm
- Carrier_Power_R: float: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEValuation:SPECTrum:AVERage
value: CalculateStruct = driver.multiEval.spectrum.average.calculate()
```

Returns the ACLR power and spectrum emission single value results of the multi-evaluation measurement. The current, average and maximum values can be retrieved. See also ‘Detailed Views: ACLR’ and ‘Detailed Views: Spectrum Emission Mask’ The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the results 1 to 18, 29 and 30 listed below. The frequency positions are only returned by FETCh and READ commands.

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

fetch(aclr_mode: *Optional*[RsCmwWcdmaMeas.enums.AclrMode] = None) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:SPECTrum:AVERage
value: ResultData = driver.multiEval.spectrum.average.fetch(aclr_mode = enums.
↪ AcclrMode.Absolute)
```

Returns the ACLR power and spectrum emission single value results of the multi-evaluation measurement. The current, average and maximum values can be retrieved. See also ‘Detailed Views: ACLR’ and ‘Detailed Views: Spectrum Emission Mask’ The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the results 1 to 18, 29 and 30 listed below. The frequency positions are only returned by FETCh and READ commands.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power Query parameter is only relevant for FETCh and READ commands. CALCulate commands return a limit check independent from the used ACLRMode.

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

read(aclr_mode: *Optional*[RsCmwWcdmaMeas.enums.AclrMode] = None) → ResultData

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:SPECtrum:AVERage
value: ResultData = driver.multiEval.spectrum.average.read(aclr_mode = enums.
    ← AcLrMode.ABSolute)
```

Returns the ACLR power and spectrum emission single value results of the multi-evaluation measurement. The current, average and maximum values can be retrieved. See also ‘Detailed Views: ACLR’ and ‘Detailed Views: Spectrum Emission Mask’ The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the results 1 to 18, 29 and 30 listed below. The frequency positions are only returned by FETCh and READ commands.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power Query parameter is only relevant for FETCh and READ commands. CALCulate commands return a limit check independent from the used ACLRMode.

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

7.4.4.2 Current

SCPI Commands

```
CALCulate:WCDma:MEASurement<Instance>:MEValuation:SPECtrum:CURRent
FETCh:WCDma:MEASurement<Instance>:MEValuation:SPECtrum:CURRent
READ:WCDma:MEASurement<Instance>:MEValuation:SPECtrum: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’
- Carrier_Power: float: float Power at the nominal carrier UL frequency Range: -100 dBm to 55 dBm, Unit: dBm
- AcLr_Minus_2: float: float Power of the adjacent channels (±1st adjacent channels at ±5 MHz from the UL frequency, ±2nd adjacent channels at ±10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- AcLr_Minus_1: float: float Power of the adjacent channels (±1st adjacent channels at ±5 MHz from the UL frequency, ±2nd adjacent channels at ±10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- AcLr_Plus_1: float: float Power of the adjacent channels (±1st adjacent channels at ±5 MHz from the UL frequency, ±2nd adjacent channels at ±10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- AcLr_Plus_2: float: float Power of the adjacent channels (±1st adjacent channels at ±5 MHz from the UL frequency, ±2nd adjacent channels at ±10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Obw: float: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- Sem_Margin_Abij: float: No parameter help available
- Sem_Margin_Bcjk: float: No parameter help available
- Sem_Margin_Cdkl: float: No parameter help available

- Sem_Margin_Efmn: float: No parameter help available
- Sem_Margin_Fenm: float: No parameter help available
- Sem_Margin_Dclk: float: No parameter help available
- Sem_Margin_Cbkj: float: No parameter help available
- Sem_Margin_Baji: float: No parameter help available
- Ue_Power: enums.ResultStatus2: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Emask_Margin_Ad: float: No parameter help available
- Emask_Margin_Da: float: No parameter help available
- Carrier_Power_L: enums.ResultStatus2: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm
- Carrier_Power_R: enums.ResultStatus2: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Carrier_Power: float: float Power at the nominal carrier UL frequency Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Obw: float: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- Sem_Margin_Abij: float: No parameter help available
- Sem_Margin_Bcjk: float: No parameter help available
- Sem_Margin_Cdkl: float: No parameter help available
- Sem_Margin_Efmn: float: No parameter help available
- Sem_Margin_Fenm: float: No parameter help available
- Sem_Margin_Dclk: float: No parameter help available
- Sem_Margin_Cbkj: float: No parameter help available
- Sem_Margin_Baji: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm

- Sem_Margin_Ad: float: No parameter help available
- Sem_Margin_Da: float: No parameter help available
- Sem_Abij_At_Freq: float: No parameter help available
- Sem_Bcjk_At_Freq: float: No parameter help available
- Sem_Cdkl_At_Freq: float: No parameter help available
- Sem_Efmn_At_Freq: float: No parameter help available
- Sem_Fenm_At_Freq: float: No parameter help available
- Sem_Dclk_At_Freq: float: No parameter help available
- Sem_Cbkj_At_Freq: float: No parameter help available
- Sem_Baji_At_Freq: float: No parameter help available
- Sem_Adat_Freq: float: No parameter help available
- Sem_Da_At_Freq: float: No parameter help available
- Carrier_Power_L: float: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm
- Carrier_Power_R: float: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:SPECTrum:CURRent
value: CalculateStruct = driver.multiEval.spectrum.current.calculate()
```

Returns the ACLR power and spectrum emission single value results of the multi-evaluation measurement. The current, average and maximum values can be retrieved. See also ‘Detailed Views: ACLR’ and ‘Detailed Views: Spectrum Emission Mask’ The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the results 1 to 18, 29 and 30 listed below. The frequency positions are only returned by FETCh and READ commands.

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

fetch(aclr_mode: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None) → ResultData

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:SPECTrum:CURRent
value: ResultData = driver.multiEval.spectrum.current.fetch(aclr_mode = enums.
↪ AcclrMode.Absolute)
```

Returns the ACLR power and spectrum emission single value results of the multi-evaluation measurement. The current, average and maximum values can be retrieved. See also ‘Detailed Views: ACLR’ and ‘Detailed Views: Spectrum Emission Mask’ The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the results 1 to 18, 29 and 30 listed below. The frequency positions are only returned by FETCh and READ commands.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power Query parameter is only relevant for FETCh and READ commands. CALCulate commands return a limit check independent from the used ACLRMode.

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

read(aclr_mode: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None) → ResultData

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:SPECTRUM:CURRENT
value: ResultData = driver.multiEval.spectrum.current.read(aclr_mode = enums.
    AclrMode.Absolute)
```

Returns the ACLR power and spectrum emission single value results of the multi-evaluation measurement. The current, average and maximum values can be retrieved. See also ‘Detailed Views: ACLR’ and ‘Detailed Views: Spectrum Emission Mask’ The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the results 1 to 18, 29 and 30 listed below. The frequency positions are only returned by FETCh and READ commands.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power Query parameter is only relevant for FETCh and READ commands. CALCulate commands return a limit check independent from the used ACLRMode.

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

7.4.4.3 Maximum

SCPI Commands

```
CALCulate:WCDMA:MEASUREMENT<Instance>:MEVALUATION:SPECTRUM:MAXIMUM
FETCh:WCDMA:MEASUREMENT<Instance>:MEVALUATION:SPECTRUM:MAXIMUM
READ:WCDMA:MEASUREMENT<Instance>:MEVALUATION:SPECTRUM: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’
- Carrier_Power: float: float Power at the nominal carrier UL frequency Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_2: float: float Power of the adjacent channels (±1st adjacent channels at ±5 MHz from the UL frequency, ±2nd adjacent channels at ±10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_1: float: float Power of the adjacent channels (±1st adjacent channels at ±5 MHz from the UL frequency, ±2nd adjacent channels at ±10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_1: float: float Power of the adjacent channels (±1st adjacent channels at ±5 MHz from the UL frequency, ±2nd adjacent channels at ±10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_2: float: float Power of the adjacent channels (±1st adjacent channels at ±5 MHz from the UL frequency, ±2nd adjacent channels at ±10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Obw: float: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- Sem_Margin_Abij: float: No parameter help available

- Sem_Margin_Bcjk: float: No parameter help available
- Sem_Margin_Cdkl: float: No parameter help available
- Sem_Margin_Efmn: float: No parameter help available
- Sem_Margin_Fenm: float: No parameter help available
- Sem_Margin_Dclk: float: No parameter help available
- Sem_Margin_Cbkj: float: No parameter help available
- Sem_Margin_Baji: float: No parameter help available
- Ue_Power: enums.ResultStatus2: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Emask_Margin_Ad: float: No parameter help available
- Emask_Margin_Da: float: No parameter help available
- Carrier_Power_L: enums.ResultStatus2: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm
- Carrier_Power_R: enums.ResultStatus2: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Carrier_Power: float: float Power at the nominal carrier UL frequency Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Obw: float: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- Sem_Margin_Abij: float: No parameter help available
- Sem_Margin_Bcjk: float: No parameter help available
- Sem_Margin_Cdkl: float: No parameter help available
- Sem_Margin_Efmn: float: No parameter help available
- Sem_Margin_Fenm: float: No parameter help available
- Sem_Margin_Dclk: float: No parameter help available
- Sem_Margin_Cbkj: float: No parameter help available

- Sem_Margin_Baji: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Sem_Margin_Ad: float: No parameter help available
- Sem_Margin_Da: float: No parameter help available
- Sem_Abij_At_Freq: float: No parameter help available
- Sem_Bcjk_At_Freq: float: No parameter help available
- Sem_Cdkl_At_Freq: float: No parameter help available
- Sem_Efmn_At_Freq: float: No parameter help available
- Sem_Fenm_At_Freq: float: No parameter help available
- Sem_Dclk_At_Freq: float: No parameter help available
- Sem_Cbkj_At_Freq: float: No parameter help available
- Sem_Baji_At_Freq: float: No parameter help available
- Sem_Adat_Freq: float: No parameter help available
- Sem_Da_At_Freq: float: No parameter help available
- Carrier_Power_L: float: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm
- Carrier_Power_R: float: float Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection Range: -90 dBm to 0 dBm, Unit: dBm

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEvaluation:SPECTrum:MAXimum
value: CalculateStruct = driver.multiEval.spectrum.maximum.calculate()
```

Returns the ACLR power and spectrum emission single value results of the multi-evaluation measurement. The current, average and maximum values can be retrieved. See also ‘Detailed Views: ACLR’ and ‘Detailed Views: Spectrum Emission Mask’ The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the results 1 to 18, 29 and 30 listed below. The frequency positions are only returned by FETCh and READ commands.

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

fetch(aclr_mode: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None) → ResultData

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:SPECTrum:MAXimum
value: ResultData = driver.multiEval.spectrum.maximum.fetch(aclr_mode = enums.
    ↪AclrMode.Absolute)
```

Returns the ACLR power and spectrum emission single value results of the multi-evaluation measurement. The current, average and maximum values can be retrieved. See also ‘Detailed Views: ACLR’ and ‘Detailed Views: Spectrum Emission Mask’ The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the results 1 to 18, 29 and 30 listed below. The frequency positions are only returned by FETCh and READ commands.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as
absolute value RELative: ACLR power displayed in dB relative to carrier power Query

parameter is only relevant for FETCh and READ commands. CALCulate commands return a limit check independent from the used ACLRMode.

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

read(*aclr_mode*: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:SPECTrum:MAXimum
value: ResultData = driver.multiEval.spectrum.maximum.read(aclr_mode = enums.
↳ AclrMode.ABSolute)
```

Returns the ACLR power and spectrum emission single value results of the multi-evaluation measurement. The current, average and maximum values can be retrieved. See also ‘Detailed Views: ACLR’ and ‘Detailed Views: Spectrum Emission Mask’ The return values described below are returned by FETCh and READ commands. CALCulate commands return limit check results instead, one value for each of the results 1 to 18, 29 and 30 listed below. The frequency positions are only returned by FETCh and READ commands.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power Query parameter is only relevant for FETCh and READ commands. CALCulate commands return a limit check independent from the used ACLRMode.

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

7.4.5 Modulation

class Modulation

Modulation 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.modulation.clone()
```

Subgroups

7.4.5.1 Uephd

SCPI Commands

```
CALCulate:WCDMa:MEASurement<Instance>:MEvaluation:MODulation:UEPHd
READ:WCDMa:MEASurement<Instance>:MEvaluation:MODulation:UEPHd
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:MODulation:UEPHd
```

class Uephd

Uephd commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’

- Overall_Max_Ph_D: float: float Overall maximum phase discontinuity Range: -180 deg to 180 deg, Unit: deg
- Overall_Min_Dist: float: decimal Overall minimum slot distance between two results exceeding the dynamic limit Unit: slots
- Count_Upper_Limit: float: decimal Number of results exceeding the upper limit Range: 0 to 99999999
- Count_Dyn_Limit: float: decimal Number of results exceeding the dynamic limit Range: 0 to 99999999

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Overall_Max_Ph_D: float: float Overall maximum phase discontinuity Range: -180 deg to 180 deg, Unit: deg
- Overall_Min_Dist: int: decimal Overall minimum slot distance between two results exceeding the dynamic limit Unit: slots
- Count_Upper_Limit: int: decimal Number of results exceeding the upper limit Range: 0 to 99999999
- Count_Dyn_Limit: int: decimal Number of results exceeding the dynamic limit Range: 0 to 99999999

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:MODulation:UEPHd
value: CalculateStruct = driver.multiEval.modulation.uephd.calculate()
```

Returns the UE phase discontinuity single value results for signals without HSPA channels. The results depend on the upper limit and the dynamic limit, see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.phd. See also 'Detailed Views: Phase Discontinuity' 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:WCDma:MEASurement<instance>:MEValuation:MODulation:UEPHd
value: ResultData = driver.multiEval.modulation.uephd.fetch()
```

Returns the UE phase discontinuity single value results for signals without HSPA channels. The results depend on the upper limit and the dynamic limit, see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.phd. See also 'Detailed Views: Phase Discontinuity' 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:WCDma:MEASurement<instance>:MEValuation:MODulation:UEPHd
value: ResultData = driver.multiEval.modulation.uephd.read()
```

Returns the UE phase discontinuity single value results for signals without HSPA channels. The results depend on the upper limit and the dynamic limit, see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.phd. See also ‘Detailed Views: Phase Discontinuity’ 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 PhDhsDpcch

SCPI Commands

```
CALCulate:WCDma:MEASurement<Instance>:MEValuation:MODulation:PHDHsdpcch
READ:WCDma:MEASurement<Instance>:MEValuation:MODulation:PHDHsdpcch
FETCh:WCDma:MEASurement<Instance>:MEValuation:MODulation:PHDHsdpcch
```

class PhDhsDpcch

PhDhsDpcch commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Overall_Max_Ph_D: float: No parameter help available
- Measure_Points: float: No parameter help available
- Count_Dyn_Limit: float: decimal Number of results exceeding the limit Range: 0 to 99999999
- Ratio_Dyn_Limit: float: float Percentage of results exceeding the limit Range: 0 % to 100 %, Unit: %
- Meas_Point_Acurr: float: No parameter help available
- Meas_Point_Amax: float: No parameter help available
- Meas_Point_Bcurr: float: No parameter help available
- Meas_Point_Bmax: float: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Overall_Max_Ph_D: float: No parameter help available
- Measure_Points: int: No parameter help available
- Count_Dyn_Limit: int: decimal Number of results exceeding the limit Range: 0 to 99999999
- Ratio_Dyn_Limit: float: float Percentage of results exceeding the limit Range: 0 % to 100 %, Unit: %
- Meas_Point_Acurr: float: No parameter help available
- Meas_Point_Amax: float: No parameter help available
- Meas_Point_Bcurr: float: No parameter help available
- Meas_Point_Bmax: float: No parameter help available

calculate() → CalculateStruct


```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:MODulation:PHDHsdpcch
value: CalculateStruct = driver.multiEval.modulation.phDhsDpcch.calculate()
```

Returns the phase discontinuity HS-DPCCH single value results for signals with HS-DPCCH. The results depend on the dynamic limit and points A and B (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.phsDpcch). See also ‘Detailed Views: Phase Discontinuity’. 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:WCDma:MEASurement<instance>:MEValuation:MODulation:PHDHsdpcch
value: ResultData = driver.multiEval.modulation.phDhsDpcch.fetch()
```

Returns the phase discontinuity HS-DPCCH single value results for signals with HS-DPCCH. The results depend on the dynamic limit and points A and B (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.phsDpcch). See also ‘Detailed Views: Phase Discontinuity’. 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:WCDma:MEASurement<instance>:MEValuation:MODulation:PHDHsdpcch
value: ResultData = driver.multiEval.modulation.phDhsDpcch.read()
```

Returns the phase discontinuity HS-DPCCH single value results for signals with HS-DPCCH. The results depend on the dynamic limit and points A and B (see method RsCmwWcdmaMeas.Configure.MultiEval.Limit.phsDpcch). See also ‘Detailed Views: Phase Discontinuity’. 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 Ber

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:MEValuation:BER
FETCh:WCDma:MEASurement<Instance>:MEValuation:BER
```

class Ber

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Ber: float: float Percentage of received data bits that were erroneous Range: 0 % to 100 %, Unit: %

- Bler: float: float Percentage of received transport data blocks containing at least one erroneous bit
Range: 0 % to 100 %, Unit: %

fetch() → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:BER  
value: ResultData = driver.multiEval.ber.fetch()
```

Returns the bit error rate and the block error ratio.

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

read() → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:BER  
value: ResultData = driver.multiEval.ber.read()
```

Returns the bit error rate and the block error ratio.

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

7.4.7 Pcde

class Pcde

Pcde 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.multiEval.pcde.clone()
```

Subgroups

7.4.7.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:PCDE:CURRent  
FETCh:WCDMa:MEASurement<Instance>:MEValuation:PCDE: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'
- Pcd_Error: float: float Peak code domain error Range: -100 dB to 0 dB, Unit: dB
- Pcd_Error_Phase: enums.PcdErrorPhase: IPHase | QPHase Phase where the peak code domain error was measured IPHase: I-Signal QPHase: Q-Signal

- Pcd_Error_Code_Nr: int: decimal Code number for which the PCDE was measured Range: 0 to 255

fetch() → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:PCDE:CURRent
value: ResultData = driver.multiEval.pcde.current.fetch()
```

Returns the peak code domain error (PCDE) results. In addition to the current PCDE value, the maximum PCDE value can be retrieved. See also ‘Detailed Views: CD Monitor’

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

read() → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:PCDE:CURRent
value: ResultData = driver.multiEval.pcde.current.read()
```

Returns the peak code domain error (PCDE) results. In addition to the current PCDE value, the maximum PCDE value can be retrieved. See also ‘Detailed Views: CD Monitor’

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

7.4.7.2 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:MEValuation:PCDE:MAXimum
FETCh:WCDMa:MEASurement<Instance>:MEValuation:PCDE:MAXimum
```

class Maximum

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

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Pcd_Error: float: float Peak code domain error Range: -100 dB to 0 dB, Unit: dB
- Pcd_Error_Phase: enums.PcdErrorPhase: IPHase | QPHase Phase where the peak code domain error was measured IPHase: I-Signal QPHase: Q-Signal
- Pcd_Error_Code_Nr: int: decimal Code number for which the PCDE was measured Range: 0 to 255

fetch() → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:PCDE:MAXimum
value: ResultData = driver.multiEval.pcde.maximum.fetch()
```

Returns the peak code domain error (PCDE) results. In addition to the current PCDE value, the maximum PCDE value can be retrieved. See also ‘Detailed Views: CD Monitor’

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

read() → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:PCDE:MAXimum
value: ResultData = driver.multiEval.pcde.maximum.read()
```

Returns the peak code domain error (PCDE) results. In addition to the current PCDE value, the maximum PCDE value can be retrieved. See also ‘Detailed Views: CD Monitor’

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

7.4.8 ListPy

class ListPy

ListPy commands group definition. 181 total commands, 9 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.1 Sreliability

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SREliability
```

class Sreliability

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

fetch() → List[int]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:SREliability
value: List[int] = driver.multiEval.listPy.sreliability.fetch()
```

Returns the segment reliability for all measured 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.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return seg_reliability: decimal Comma-separated list of values, one per measured segment The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.

7.4.8.2 UePower

class UePower

UePower 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.multiEval.listPy.uePower.clone()
```

Subgroups

7.4.8.2.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:UEPower:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:UEPower:CURRent
value: List[float] = driver.multiEval.listPy.uePower.current.fetch()
```

Returns the UE power vs. slot results in list mode.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float User equipment power, one value per slot. The list contains results for all active segments (segments for which any measurement has been enabled) . If another measurement has been enabled for a segment, but the UE power vs. slot measurement is disabled, NCAPs are returned for that segment. Example: segment 1 with 10 slots active, segment 2 with 50 slots inactive, segment 3 with 12 slots active. 22 power results are returned. Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.3 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. 20 total commands, 7 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

7.4.8.3.1 UePower

class UePower

UePower 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.multiEval.listPy.segment.uePower.clone()
```

Subgroups

7.4.8.3.1.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:UEPower:CURRENT
```

class Current

Current 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Ue_Power: List[float]: float User equipment power, one value per slot. The list contains results for the indicated segment no. If another measurement has been enabled for a segment, but the UE power vs. slot measurement is disabled, NCAP is returned. Range: -100 dBm to 55 dBm, Unit: dBm

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:UEPower:CURRENT
value: FetchStruct = driver.multiEval.listPy.segment.uePower.current.
↳fetch(segment = repcap.Segment.Default)
```

Returns the UE power vs. slot results for segment <no> in 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 FetchStruct structure arguments.

7.4.8.3.2 Phd

class Phd

Phd 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.multiEval.listPy.segment.phd.clone()
```

Subgroups

7.4.8.3.2.1 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:PHD:CURRent
```

class Current

Current 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Phd: List[float]: float Comma-separated list of phase discontinuity results, one value per slot. The list contains results for the indicated segment no. If another measurement has been enabled for a segment, but the phase discontinuity measurement is disabled, NCAPs are returned for that segment. Range: -180 deg to 180 deg, Unit: deg

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:PHD:CURRent
value: FetchStruct = driver.multiEval.listPy.segment.phd.current.fetch(segment_
↳= repcap.Segment.Default)
```

Returns the phase discontinuity vs. slot results for segment <no> in list mode. Each value indicates the phase discontinuity at the boundary between the slot and the previous slot. If the slot or the previous slot is not measured, NCAP is returned.

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.

7.4.8.3.3 Pcde

class Pcde

Pcde commands group definition. 2 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.segment.pcde.clone()
```

Subgroups

7.4.8.3.3.1 Current

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:PCDE:CURRent
```

class Current

Current 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Pcd_Error: float: float Peak code domain error Range: -100 dB to 0 dB, Unit: dB
- Pcd_Error_Phase: enums.PcdErrorPhase: No parameter help available
- Pcd_Error_Code_Nr: int: No parameter help available

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:PCDE:CURRent
value: FetchStruct = driver.multiEval.listPy.segment.pcde.current.fetch(segment_
↳= repcap.Segment.Default)
```

Returns the peak code domain error (PCDE) results for segment <no> in 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 FetchStruct structure arguments.

7.4.8.3.3.2 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:PCDE:MAXimum
```

class Maximum

Maximum 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Pcd_Error: float: float Peak code domain error Range: -100 dB to 0 dB, Unit: dB
- Pcd_Error_Phase: enums.PcdErrorPhase: No parameter help available
- Pcd_Error_Code_Nr: int: No parameter help available

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:PCDE:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.pcde.maximum.fetch(segment_
↳= repcap.Segment.Default)
```

Returns the peak code domain error (PCDE) results for segment <no> in 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 FetchStruct structure arguments.

7.4.8.3.4 CdPower

class CdPower

CdPower commands group definition. 5 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.cdPower.clone()
```

Subgroups

7.4.8.3.4.1 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CDPower:CURRent
```

class Current

Current 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:CDPower:CURRent
value: FetchStruct = driver.multiEval.listPy.segment.cdPower.current.
↳fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP vs. slot results for segment <no> in list mode. 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.

7.4.8.3.4.2 Average

SCPI Commands

FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CDPower:AVERage

class Average

Average 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳ :CDPower:AVERage
value: FetchStruct = driver.multiEval.listPy.segment.cdPower.average.
↳ fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP vs. slot results for segment <no> in list mode. 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.

7.4.8.3.4.3 Minimum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CDPower:MINimum
```

class Minimum

Minimum 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳ :CDPower:MINimum
value: FetchStruct = driver.multiEval.listPy.segment.cdPower.minimum.
↳ fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP vs. slot results for segment <no> in list mode. 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.

7.4.8.3.4.4 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CDPower:MAXimum
```

class Maximum

Maximum 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳ :CDPower:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.cdPower.maximum.
↳ fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP vs. slot results for segment <no> in list mode. 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.

7.4.8.3.4.5 StandardDev

SCPI Commands

FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CDPower:SDEViation

class StandardDev

StandardDev 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:CDPower:SDEViation
value: FetchStruct = driver.multiEval.listPy.segment.cdPower.standardDev.
↳fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP vs. slot results for segment <no> in list mode. 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.

7.4.8.3.5 Spectrum

class Spectrum

Spectrum commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.3.5.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:SPECtrum:CURRent
```

class Current

Current 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.
- **Return_Code:** int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- **Carrier_Power:** float: float Power at the nominal carrier UL frequency Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Minus_2:** float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Minus_1:** float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Plus_1:** float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm

- **Aclr_Plus_2:** float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Obw:** float: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- **Emask_Margin_Ab:** float: No parameter help available
- **Emask_Margin_Bc:** float: No parameter help available
- **Emask_Margin_Cd:** float: No parameter help available
- **Emask_Margin_Ef:** float: No parameter help available
- **Emask_Margin_Fe:** float: No parameter help available
- **Emask_Margin_Dc:** float: No parameter help available
- **Emask_Margin_Cb:** float: No parameter help available
- **Emask_Margin_Ba:** float: No parameter help available
- **Ue_Power:** float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- **Emask_Margin_Had:** float: No parameter help available
- **Emask_Margin_Hda:** float: No parameter help available

fetch(*aclr_mode*: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None, *segment*=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳ :SPECTrum:CURRent
value: FetchStruct = driver.multiEval.listPy.segment.spectrum.current.
↳ fetch(aclr_mode = enums.AclrMode.ABSolute, segment = reprcap.Segment.Default)
```

Returns the ACLR power and spectrum emission single value results for segment <no> in list mode.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

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.

7.4.8.3.5.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:SPECTrum:AVERage
```

class Average

Average 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.

- **Return_Code**: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- **Carrier_Power**: float: float Power at the nominal carrier UL frequency Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Minus_2**: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Minus_1**: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Plus_1**: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Plus_2**: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Obw**: float: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- **Emask_Margin_Ab**: float: No parameter help available
- **Emask_Margin_Bc**: float: No parameter help available
- **Emask_Margin_Cd**: float: No parameter help available
- **Emask_Margin_Ef**: float: No parameter help available
- **Emask_Margin_Fe**: float: No parameter help available
- **Emask_Margin_Dc**: float: No parameter help available
- **Emask_Margin_Cb**: float: No parameter help available
- **Emask_Margin_Ba**: float: No parameter help available
- **Ue_Power**: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- **Emask_Margin_Had**: float: No parameter help available
- **Emask_Margin_Hda**: float: No parameter help available

fetch(*aclr_mode*: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None, *segment*=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↪:SPECTrum:AVERage
value: FetchStruct = driver.multiEval.listPy.segment.spectrum.average.
↪fetch(aclr_mode = enums.AclrMode.ABSolute, segment = repcap.Segment.Default)
```

Returns the ACLR power and spectrum emission single value results for segment <no> in list mode.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

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.

7.4.8.3.5.3 Maximum

SCPI Commands

<code>FETCH:WCDMA:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:SPECtrum:MAXimum</code>

class Maximum

Maximum 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Carrier_Power: float: float Power at the nominal carrier UL frequency Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_1: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_2: float: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Obw: float: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- Emask_Margin_Ab: float: No parameter help available
- Emask_Margin_Bc: float: No parameter help available
- Emask_Margin_Cd: float: No parameter help available
- Emask_Margin_Ef: float: No parameter help available
- Emask_Margin_Fe: float: No parameter help available
- Emask_Margin_Dc: float: No parameter help available
- Emask_Margin_Cb: float: No parameter help available
- Emask_Margin_Ba: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Emask_Margin_Had: float: No parameter help available
- Emask_Margin_Hda: float: No parameter help available

fetch(*aclr_mode*: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None, *segment*=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳ :SPECTrum:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.spectrum.maximum.
↳ fetch(aclr_mode = enums.AclrMode.ABSolute, segment = repcap.Segment.Default)
```

Returns the ACLR power and spectrum emission single value results for segment <no> in list mode.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

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.

7.4.8.3.6 Modulation

class Modulation

Modulation commands group definition. 4 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.modulation.clone()
```

Subgroups

7.4.8.3.6.1 StandardDev

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:MODulation:SDEviation
```

class StandardDev

StandardDev 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.
- **Return_Code**: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- **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: %
- **Mag_Error_Rms**: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %

- **Mag_Error_Peak**: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- **Phase_Error_Rms**: float: No parameter help available
- **Phase_Error_Peak**: float: No parameter help available
- **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
- **Carrier_Freq_Err**: float: No parameter help available
- **Transmit_Time_Err**: float: No parameter help available
- **Ue_Power**: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:MODulation:SDEViation
value: FetchStruct = driver.multiEval.listPy.segment.modulation.standardDev.
↳fetch(segment = reprcap.Segment.Default)
```

Returns modulation single value results for segment <no> in 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 FetchStruct structure arguments.

7.4.8.3.6.2 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:MODulation:MAXimum
```

class Maximum

Maximum 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.
- **Return_Code**: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- **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: %
- **Mag_Error_Rms**: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- **Mag_Error_Peak**: float: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- **Phase_Error_Rms**: float: No parameter help available

- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDMA: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.

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.

7.4.8.3.6.3 Current

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:LIST:SEGMENT<Segment>:MODULATION:CURRENT
```

class Current

Current 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- 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: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERAGE: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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

- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDMA: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.

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.

7.4.8.3.6.4 Average

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:LIST:SEGMENT<Segment>:MODULATION:AVERAGE
```

class Average

Average 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- 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: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 % (AVERAGE: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDMA: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.

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.

7.4.8.3.7 CdError

class CdError

CdError commands group definition. 4 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.cdError.clone()
```

Subgroups

7.4.8.3.7.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGment<Segment>:CDError:CURRENT
```

class Current

Current 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.
- **Return_Code:** int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- **Dpcch:** float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Dpdch:** float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Hsdpcch:** float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Edpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↪:CDERror:CURRent
value: FetchStruct = driver.multiEval.listPy.segment.cdError.current.
↪fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDE vs. slot results for segment <no> in 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 FetchStruct structure arguments.

7.4.8.3.7.2 Average

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CDERror:AVERage
```

class Average

Average 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Edpdch_1: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SEGMENT<nr>
↪:CDERROR:AVERAGE
value: FetchStruct = driver.multiEval.listPy.segment.cdError.average.
↪fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDE vs. slot results for segment <no> in 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 FetchStruct structure arguments.

7.4.8.3.7.3 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:LIST:SEGMENT<Segment>:CDERROR:MAXIMUM
```

class Maximum

Maximum 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB

- Edpdch_2: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↪:CDERror:MAXimum
value: FetchStruct = driver.multiEval.listPy.segment.cdError.maximum.
↪fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDE vs. slot results for segment <no> in 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 FetchStruct structure arguments.

7.4.8.3.7.4 StandardDev

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<Segment>:CDERror:SDEVIation
```

class StandardDev

StandardDev 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.
- Return_Code: int: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Dpdch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpcch: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVIation 0 dB to 50 dB) , Unit: dB

- Edpdch_3: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVI-ation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: float: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEVI-ation 0 dB to 50 dB) , Unit: dB

fetch(segment=<Segment.Default: -1>) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↪:CDError:SDEVIation
value: FetchStruct = driver.multiEval.listPy.segment.cdError.standardDev.
↪fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDE vs. slot results for segment <no> in 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 FetchStruct structure arguments.

7.4.8.4 Phd

class Phd

Phd 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.multiEval.listPy.phd.clone()
```

Subgroups

7.4.8.4.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:PHD:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PHD:CURRent
value: List[float] = driver.multiEval.listPy.phd.current.fetch()
```

Returns the phase discontinuity vs. slot results in list mode. Each value indicates the phase discontinuity at the boundary between the slot and the previous slot. If the slot or the previous slot is not measured, NCAP is returned.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phd: float Comma-separated list of phase discontinuity results, one value per slot. The list contains results for all active segments (segments for which any measurement has been enabled) . If another measurement has been enabled for a segment, but the phase discontinuity measurement is disabled, NCAPs are returned for that segment. Example: segment 1 with 10 slots active, segment 2 with 50 slots inactive, segment 3 with 12 slots active. 22 phase discontinuity results are returned. Range: -180 deg to 180 deg, Unit: deg

7.4.8.5 Pcde

class Pcde

Pcde commands group definition. 8 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.pcde.clone()
```

Subgroups

7.4.8.5.1 Code

class Code

Code commands group definition. 2 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.pcde.code.clone()
```

Subgroups

7.4.8.5.1.1 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:PCDE:CODE:MAXimum
```

class Maximum

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

fetch() → List[int]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:PCDE:CODE:MAXimum
value: List[int] = driver.multiEval.listPy.pcde.code.maximum.fetch()
```

Return the code number for which the peak code domain error was measured, for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return pcd_error_code_nr: decimal Comma-separated list of values, one per measured segment Range: 0 to 255

7.4.8.5.1.2 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:PCDE:CODE:CURRENT
```

class Current

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

fetch() → List[int]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:CODE:CURRENT
value: List[int] = driver.multiEval.listPy.pcde.code.current.fetch()
```

Return the code number for which the peak code domain error was measured, for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return pcd_error_code_nr: decimal Comma-separated list of values, one per measured segment Range: 0 to 255

7.4.8.5.2 Phase

class Phase

Phase commands group definition. 2 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.pcde.phase.clone()
```

Subgroups

7.4.8.5.2.1 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:PCDE:PHASe:MAXimum
```

class Maximum

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

fetch() → List[RsCmwWcdmaMeas.enums.PcdErrorPhase]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:PHASe:MAXimum
value: List[enums.PcdErrorPhase] = driver.multiEval.listPy.pcde.phase.maximum.
↪ fetch()
```

Return the phase where the peak code domain error was measured, for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return pcd_error_phase: IPHase | QPHase Comma-separated list of values, one per measured segment
IPHase: I-Signal QPHase: Q-Signal

7.4.8.5.2.2 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:LIST:PCDE:PHASe:CURRent
```

class Current

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

fetch() → List[RsCmwWcdmaMeas.enums.PcdErrorPhase]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:PHASe:CURRent
value: List[enums.PcdErrorPhase] = driver.multiEval.listPy.pcde.phase.current.
↪ fetch()
```

Return the phase where the peak code domain error was measured, for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return pcd_error_phase: IPHase | QPHase Comma-separated list of values, one per measured segment
IPHase: I-Signal QPHase: Q-Signal

7.4.8.5.3 Error

class Error

Error commands group definition. 2 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.pcde.error.clone()
```

Subgroups

7.4.8.5.3.1 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:PCDE:ERror:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:ERror:MAXimum
value: List[float] = driver.multiEval.listPy.pcde.error.maximum.fetch()
```

Return peak code domain error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return pcde_error: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.5.3.2 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:PCDE:ERror:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:ERror:CURRent
value: List[float] = driver.multiEval.listPy.pcde.error.current.fetch()
```

Return peak code domain error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return pcde_error: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.5.4 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:PCDE:CURRENT
```

class Current

Current 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Pcd_Error: List[float]: float Peak code domain error Range: -100 dB to 0 dB, Unit: dB
- Pcd_Error_Phase: List[enums.PcdErrorPhase]: No parameter help available
- Pcd_Error_Code_Nr: List[int]: No parameter help available

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:CURRENT
value: FetchStruct = driver.multiEval.listPy.pcd.current.fetch()
```

Return the peak code domain error (PCDE) results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval. ListPy.count.

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

7.4.8.5.5 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:PCDE:MAXimum
```

class Maximum

Maximum 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Pcd_Error: List[float]: float Peak code domain error Range: -100 dB to 0 dB, Unit: dB
- Pcd_Error_Phase: List[enums.PcdErrorPhase]: No parameter help available

- Pcd_Error_Code_Nr: List[int]: No parameter help available

fetch() → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:MAXimum
value: FetchStruct = driver.multiEval.listPy.pcde.maximum.fetch()
```

Return the peak code domain error (PCDE) results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval. ListPy.count.

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

7.4.8.6 CdPower

class CdPower

CdPower commands group definition. 30 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.cdPower.clone()
```

Subgroups

7.4.8.6.1 Dpcch

class Dpcch

Dpcch commands group definition. 5 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.cdPower.dpcch.clone()
```

Subgroups

7.4.8.6.1.1 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:CDPower:DPCCh:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:CDPOWER:DPCCH:CURRENt
value: List[float] = driver.multiEval.listPy.cdPower.dpcch.current.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.1.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:LIST:CDPOWER:DPCCH:AVERAge
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:CDPOWER:DPCCH:AVERAge
value: List[float] = driver.multiEval.listPy.cdPower.dpcch.average.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.1.3 Minimum

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:LIST:CDPOWER:DPCCH:MINimum
```

class Minimum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:CDPOWER:DPCCH:MINimum
value: List[float] = driver.multiEval.listPy.cdPower.dpcch.minimum.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.1.4 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:DPCCh:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cdPower.dpcch.maximum.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.1.5 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:DPCCh:SDEviation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDPower:DPCCh:SDEviation
value: List[float] = driver.multiEval.listPy.cdPower.dpcch.standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.2 Dpdch

class Dpdch

Dpdch commands group definition. 5 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.cdPower.dpdch.clone()
```

Subgroups

7.4.8.6.2.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:DPDCh:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:CURRent
value: List[float] = driver.multiEval.listPy.cdPower.dpdch.current.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpdch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.2.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:DPDCh:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:AVERage
value: List[float] = driver.multiEval.listPy.cdPower.dpdch.average.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpdch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.2.3 Minimum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:DPDCh:MINimum
```

class Minimum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:MINimum
value: List[float] = driver.multiEval.listPy.cdPower.dpdch.minimum.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpdch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.2.4 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:DPDCh:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:MAXimum
value: List[float] = driver.multiEval.listPy.cdPower.dpdch.maximum.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpdch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.2.5 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:DPDCh:SDEVIation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>
↳:MEVALUATION:LIST:CDPOWER:DPDCH:SDEViation
value: List[float] = driver.multiEval.listPy.cdPower.dpdch.standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpdch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.3 Hsdpcch

class Hsdpcch

Hsdpcch commands group definition. 5 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.cdPower.hsdpcch.clone()
```

Subgroups

7.4.8.6.3.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:LIST:CDPOWER:HSDPCCH:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>
↳:MEVALUATION:LIST:CDPOWER:HSDPCCH:CURRENT
value: List[float] = driver.multiEval.listPy.cdPower.hsdpcch.current.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return hsdpcch: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.3.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:HSDPcch:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDPower:HSDPcch:AVERage
value: List[float] = driver.multiEval.listPy.cdPower.hsdpcch.average.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return hsdpcch: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.3.3 Minimum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:HSDPcch:MINimum
```

class Minimum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDPower:HSDPcch:MINimum
value: List[float] = driver.multiEval.listPy.cdPower.hsdpcch.minimum.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return hsdpcch: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.3.4 Maximum

SCPI Commands

`FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:HSDPcch:MAXimum`

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDPower:HSDPcch:MAXimum
value: List[float] = driver.multiEval.listPy.cdPower.hsdpcch.maximum.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return hsdpcch: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.3.5 StandardDev

SCPI Commands

`FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:HSDPcch:SDEviation`

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDPower:HSDPcch:SDEviation
value: List[float] = driver.multiEval.listPy.cdPower.hsdpcch.standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return hsdpcch: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.4 Edpcch

class Edpcch

Edpcch commands group definition. 5 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.cdPower.edpcch.clone()
```

Subgroups

7.4.8.6.4.1 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDPower:EDPCch:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:CDPower:EDPCch:CURRent
value: List[float] = driver.multiEval.listPy.cdPower.edpcch.current.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return edpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.4.2 Average

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDPower:EDPCch:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:CDPower:EDPCch:AVERage
value: List[float] = driver.multiEval.listPy.cdPower.edpcch.average.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return edpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.4.3 Minimum

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:CDPower:EDPCch:MINimum`

class Minimum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:CDPower:EDPCch:MINimum
value: List[float] = driver.multiEval.listPy.cdPower.edpcch.minimum.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return edpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.4.4 Maximum

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:CDPower:EDPCch:MAXimum`

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:CDPower:EDPCch:MAXimum
value: List[float] = driver.multiEval.listPy.cdPower.edpcch.maximum.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return edpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.4.5 StandardDev

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:MEvaluation:LIST:CDPower:EDPCch:SDEviation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>
→:MEvaluation:LIST:CDPower:EDPCch:SDEviation
value: List[float] = driver.multiEval.listPy.cdPower.edpcch.standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return edpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.6.5 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.multiEval.listPy.cdPower.edpdch.repcap_edpdChannel_get()
driver.multiEval.listPy.cdPower.edpdch.repcap_edpdChannel_set(repcap.EdpdChannel.Nr1)
```

class Edpdch

Edpdch commands group definition. 5 total commands, 5 Sub-groups, 0 group commands Repeated Capability:
EdpdChannel, default value after init: EdpdChannel.Nr1

Cloning the Group

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

Subgroups

7.4.8.6.5.1 Current

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:MEvaluation:LIST:CDPower:EDPDch<EdpdChannel>:CURRENT
```

class Current

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

fetch(edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:EDPDch<nr>
↪:CURRent
value: List[float] = driver.multiEval.listPy.cdPower.edpdch.current.
↪fetch(edpdChannel = repcap.EdpdChannel.Default)
```

Return RMS CDP and CDE vs. slot values for a selected E-DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.5.2 Average

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDPower:EDPDch<EdpdChannel>:AVERage
```

class Average

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

fetch(edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:EDPDch<nr>
↪:AVERage
value: List[float] = driver.multiEval.listPy.cdPower.edpdch.average.
↪fetch(edpdChannel = repcap.EdpdChannel.Default)
```

Return RMS CDP and CDE vs. slot values for a selected E-DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.5.3 Minimum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDPower:EDPDch<EdpdChannel>:MINimum
```

class Minimum

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

fetch(edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:EDPDch<nr>
↳:MINimum
value: List[float] = driver.multiEval.listPy.cdPower.edpdch.minimum.
↳fetch(edpdChannel = repcap.EdpdChannel.Default)
```

Return RMS CDP and CDE vs. slot values for a selected E-DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.5.4 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDPower:EDPDch<EdpdChannel>:MAXimum
```

class Maximum

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

fetch(edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:EDPDch<nr>
↳:MAXimum
value: List[float] = driver.multiEval.listPy.cdPower.edpdch.maximum.
↳fetch(edpdChannel = repcap.EdpdChannel.Default)
```

Return RMS CDP and CDE vs. slot values for a selected E-DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.5.5 StandardDev

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDPower:EDPDch<EdpdChannel>:SDEviation
```

class StandardDev

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

fetch(edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:EDPDch<nr>
↪:SDEViation
value: List[float] = driver.multiEval.listPy.cdPower.edpdch.standardDev.
↪fetch(edpdChannel = repcap.EdpdChannel.Default)
```

Return RMS CDP and CDE vs. slot values for a selected E-DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.6.6 StandardDev

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDPower:SDEViation
```

class StandardDev

StandardDev 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:SDEviation
value: FetchStruct = driver.multiEval.listPy.cdPower.standardDev.fetch()
```

Return the RMS CDP vs. slot results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

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

7.4.8.6.7 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:MAXimum
```

class Maximum

Maximum 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:MAXimum
value: FetchStruct = driver.multiEval.listPy.cdPower.maximum.fetch()
```

Return the RMS CDP vs. slot results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

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

7.4.8.6.8 Minimum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:MINimum
```

class Minimum

Minimum 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:MINimum
value: FetchStruct = driver.multiEval.listPy.cdPower.minimum.fetch()
```

Return the RMS CDP vs. slot results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

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

7.4.8.6.9 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:AVERage
```

class Average

Average 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:AVERage
value: FetchStruct = driver.multiEval.listPy.cdPower.average.fetch()
```

Return the RMS CDP vs. slot results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

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

7.4.8.6.10 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDPower:CURRent
```

class Current

Current 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: List[float]: float RMS CDP values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:CURRent
value: FetchStruct = driver.multiEval.listPy.cdPower.current.fetch()
```

Return the RMS CDP vs. slot results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

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

7.4.8.7 Spectrum

class Spectrum

Spectrum commands group definition. 48 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.spectrum.clone()
```

Subgroups

7.4.8.7.1 UePower

class UePower

UePower commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.1.1 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SPECTrum:UEPower:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪ :MEValuation:LIST:SPECTrum:UEPower:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.uePower.current.fetch()
```

Return the UE power for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.1.2 Average

SCPI Commands

`FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:UEPower:AVERage`

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:UEPower:AVERage
value: List[float] = driver.multiEval.listPy.spectrum.uePower.average.fetch()
```

Return the UE power for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.1.3 Maximum

SCPI Commands

`FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:UEPower:MAXimum`

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:UEPower:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.uePower.maximum.fetch()
```

Return the UE power for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.2 Emask

class Emask

Emask commands group definition. 30 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.spectrum.emask.clone()
```

Subgroups

7.4.8.7.2.1 Hda

class Hda

Hda commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.2 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:HDA:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:HDA:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.emask.hda.current.fetch()
```

Return the limit line margin values for limit line H for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -130 dB to 130 dB, Unit: dB

7.4.8.7.2.3 Average

SCPI Commands

`FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SPECtrum:EMASk:HDA:AVERAge`

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:SPECtrum:EMASk:HDA:AVERAge
value: List[float] = driver.multiEval.listPy.spectrum.emask.hda.average.fetch()
```

Return the limit line margin values for limit line H for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -130 dB to 130 dB, Unit: dB

7.4.8.7.2.4 Maximum

SCPI Commands

`FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SPECtrum:EMASk:HDA:MAXimum`

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:SPECtrum:EMASk:HDA:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.hda.maximum.fetch()
```

Return the limit line margin values for limit line H for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -130 dB to 130 dB, Unit: dB

7.4.8.7.2.5 Had

class Had

Had commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.6 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:HAD:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:HAD:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.emask.had.current.fetch()
```

Return the limit line margin values for limit line H for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -130 dB to 130 dB, Unit: dB

7.4.8.7.2.7 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:HAD:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:HAD:AVERage
value: List[float] = driver.multiEval.listPy.spectrum.emask.had.average.fetch()
```

Return the limit line margin values for limit line H for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -130 dB to 130 dB, Unit: dB

7.4.8.7.2.8 Maximum

SCPI Commands

`FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SPECTrum:EMASk:HAD:MAXimum`

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:SPECTrum:EMASk:HAD:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.had.maximum.fetch()
```

Return the limit line margin values for limit line H for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -130 dB to 130 dB, Unit: dB

7.4.8.7.2.9 Ab

class Ab

Ab commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.10 Current

SCPI Commands


```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:AB:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:AB:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.emask.ab.current.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.11 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:AB:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:AB:AVERage
value: List[float] = driver.multiEval.listPy.spectrum.emask.ab.average.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.12 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:AB:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:AB:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.ab.maximum.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.13 Bc

class Bc

Bc commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.14 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:BC:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:BC:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.emask.bc.current.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.15 Average

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:SPECTrum:EMASk:BC:AVERAge
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:SPECTrum:EMASk:BC:AVERAge
value: List[float] = driver.multiEval.listPy.spectrum.emask.bc.average.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.16 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:SPECTrum:EMASk:BC:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:SPECTrum:EMASk:BC:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.bc.maximum.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.17 Cd

class Cd

Cd commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.18 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:CD:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:CD:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.emask.cd.current.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.19 Average

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:CD:AVERAge
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:CD:AVERAge
value: List[float] = driver.multiEval.listPy.spectrum.emask.cd.average.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.20 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:CD:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:CD:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.cd.maximum.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.21 Ef

class Ef

Ef commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.22 Current

SCPI Commands

`FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:EF:CURRent`

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:EF:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.emask.ef.current.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.23 Average

SCPI Commands

`FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:EF:AVERAge`

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:EF:AVERAge
value: List[float] = driver.multiEval.listPy.spectrum.emask.ef.average.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.24 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:SPECtrum:EMASk:EF:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:SPECtrum:EMASk:EF:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.ef.maximum.fetch()
```

Return the limit line margin values in the 4 emission mask areas below the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.25 Fe

class Fe

Fe commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.26 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:SPECtrum:EMASk:FE:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:SPECtrum:EMASk:FE:CURRENT
value: List[float] = driver.multiEval.listPy.spectrum.emask.fe.current.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.27 Average

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:FE:AVERage`

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:FE:AVERage
value: List[float] = driver.multiEval.listPy.spectrum.emask.fe.average.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.28 Maximum

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:FE:MAXimum`

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:FE:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.fe.maximum.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
 Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.29 Dc

class Dc

Dc commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.30 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:DC:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:DC:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.emask.dc.current.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
 Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.31 Average

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:DC:AVERAge
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:DC:AVERAge
value: List[float] = driver.multiEval.listPy.spectrum.emask.dc.average.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.32 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:DC:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:DC:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.dc.maximum.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.33 Cb

class Cb

Cb commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.34 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEValuation:LIST:SPECTrum:EMASk:CB:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEValuation:LIST:SPECTrum:EMASk:CB:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.emask.cb.current.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.35 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEValuation:LIST:SPECTrum:EMASk:CB:AVERAge
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEValuation:LIST:SPECTrum:EMASk:CB:AVERAge
value: List[float] = driver.multiEval.listPy.spectrum.emask.cb.average.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.36 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:CB:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:CB:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.cb.maximum.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.37 Ba

class Ba

Ba commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.2.38 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASk:BA:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:BA:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.emask.ba.current.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.39 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASK:BA:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASK:BA:AVERage
value: List[float] = driver.multiEval.listPy.spectrum.emask.ba.average.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.2.40 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:EMASK:BA:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASK:BA:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.emask.ba.maximum.fetch()
```

Return the limit line margin values in the 4 emission mask areas above the carrier frequency for all measured list mode segments. A positive result indicates that the trace is located above the limit line, i.e. the limit is exceeded.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return emask_margin: float Comma-separated list of values, one per measured segment
Range: -100 dB to 90 dB, Unit: dB

7.4.8.7.3 Cpower

class Cpower

Cpower commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.3.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:CPOWer:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:CPOWer:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.cpower.current.fetch()
```

Return the power at the nominal carrier frequency for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return carrier_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.3.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:CPOWer:AVERAge
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:CPOWer:AVERage
value: List[float] = driver.multiEval.listPy.spectrum.cpower.average.fetch()
```

Return the power at the nominal carrier frequency for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return carrier_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.3.3 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:CPOWer:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:CPOWer:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.cpower.maximum.fetch()
```

Return the power at the nominal carrier frequency for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return carrier_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.4 Aclr

class Aclr

Aclr 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.listPy.spectrum.aclr.clone()
```

Subgroups

7.4.8.7.4.1 M<Minus>

RepCap Settings

```
# Range: Ch1 .. Ch2
rc = driver.multiEval.listPy.spectrum.aclr.m.repcap_minus_get()
driver.multiEval.listPy.spectrum.aclr.m.repcap_minus_set(repcap.Minus.Ch1)
```

class M

M commands group definition. 3 total commands, 3 Sub-groups, 0 group commands Repeated Capability: Minus, default value after init: Minus.Ch1

Cloning the Group

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

Subgroups

7.4.8.7.4.2 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:ACLR:M<Minus>:CURRENT
```

class Current

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

fetch(aclr_mode: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None, minus=<Minus.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:M<nr>
↪:CURRENT
value: List[float] = driver.multiEval.listPy.spectrum.aclr.m.current.fetch(aclr_
↪mode = enums.AclrMode.ABSolute, minus = repcap.Minus.Default)
```

Return the power of the adjacent channels for all measured list mode segments. IN-TRO_CMD_HELP: The adjacent channel selected via M<no>/P<no> is at the following frequency relative to the carrier frequency:

- M1 = -5 MHz, M2 = -10 MHz
- P1 = +5 MHz, P2 = +10 MHz

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

param minus optional repeated capability selector. Default value: Ch1 (settable in the interface 'M')

return aclr: float Comma-separated list of values, one per measured segment Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.4.3 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:ACLR:M<Minus>:AVERage
```

class Average

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

fetch(aclr_mode: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None, minus=<Minus.Default: -1>) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:M<nr>
↳:AVERage
value: List[float] = driver.multiEval.listPy.spectrum.aclr.m.average.fetch(aclr_
↳mode = enums.AclrMode.ABSolute, minus = repcap.Minus.Default)
```

Return the power of the adjacent channels for all measured list mode segments. IN-

TRO_CMD_HELP: The adjacent channel selected via M<no>/P<no> is at the following frequency relative to the carrier frequency:

- M1 = -5 MHz, M2 = -10 MHz
- P1 = +5 MHz, P2 = +10 MHz

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

param minus optional repeated capability selector. Default value: Ch1 (settable in the interface 'M')

return aclr: float Comma-separated list of values, one per measured segment Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.4.4 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:ACLR:M<Minus>:MAXimum
```

class Maximum

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

fetch(aclr_mode: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None, minus=<Minus.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:M<nr>
↪:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.aclr.m.maximum.fetch(aclr_
↪mode = enums.AclrMode.ABSolute, minus = repcap.Minus.Default)
```

Return the power of the adjacent channels for all measured list mode segments. IN-

TRO_CMD_HELP: The adjacent channel selected via M<no>/P<no> is at the following frequency relative to the carrier frequency:

- M1 = -5 MHz, M2 = -10 MHz
- P1 = +5 MHz, P2 = +10 MHz

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

param minus optional repeated capability selector. Default value: Ch1 (settable in the interface 'M')

return aclr: float Comma-separated list of values, one per measured segment Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.4.5 P<Plus>

RepCap Settings

```
# Range: Ch1 .. Ch2
rc = driver.multiEval.listPy.spectrum.aclr.p.repcap_plus_get()
driver.multiEval.listPy.spectrum.aclr.p.repcap_plus_set(repcap.Plus.Ch1)
```

class P

P commands group definition. 3 total commands, 3 Sub-groups, 0 group commands Repeated Capability: Plus, default value after init: Plus.Ch1

Cloning the Group

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

Subgroups

7.4.8.7.4.6 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:ACLR:P<Plus>:CURRent
```

class Current

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

fetch(*aclr_mode*: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None, *plus*=<Plus.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:P<nr>
↪:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.aclr.p.current.fetch(aclr_
↪mode = enums.AclrMode.ABSolute, plus = repcap.Plus.Default)
```

Return the power of the adjacent channels for all measured list mode segments. IN-TRO_CMD_HELP: The adjacent channel selected via M<no>/P<no> is at the following frequency relative to the carrier frequency:

- M1 = -5 MHz, M2 = -10 MHz
- P1 = +5 MHz, P2 = +10 MHz

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

param plus optional repeated capability selector. Default value: Ch1 (settable in the interface 'P')

return aclr: float Comma-separated list of values, one per measured segment Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.4.7 Average

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:ACLR:P<Plus>:AVERage
```

class Average

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

fetch(*aclr_mode*: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None, *plus*=<Plus.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:P<nr>
↪:AVERage
value: List[float] = driver.multiEval.listPy.spectrum.aclr.p.average.fetch(aclr_
↪mode = enums.AclrMode.ABSolute, plus = repcap.Plus.Default)
```

Return the power of the adjacent channels for all measured list mode segments. IN-TRO_CMD_HELP: The adjacent channel selected via M<no>/P<no> is at the following frequency relative to the carrier frequency:

- M1 = -5 MHz, M2 = -10 MHz
- P1 = +5 MHz, P2 = +10 MHz

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

param plus optional repeated capability selector. Default value: Ch1 (settable in the interface 'P')

return aclr: float Comma-separated list of values, one per measured segment Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.4.8 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:SPECTrum:ACLR:P<Plus>:MAXimum
```

class Maximum

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

fetch(aclr_mode: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None, plus=<Plus.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SPECTrum:ACLR:P<nr>
↳:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.aclr.p.maximum.fetch(aclr_
↳mode = enums.AclrMode.ABSolute, plus = repcap.Plus.Default)
```

Return the power of the adjacent channels for all measured list mode segments. IN-

TRO_CMD_HELP: The adjacent channel selected via M<no>/P<no> is at the following frequency relative to the carrier frequency:

- M1 = -5 MHz, M2 = -10 MHz
- P1 = +5 MHz, P2 = +10 MHz

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

param plus optional repeated capability selector. Default value: Ch1 (settable in the interface 'P')

return aclr: float Comma-separated list of values, one per measured segment Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.7.5 Obw

class Obw

Obw commands group definition. 3 total commands, 3 Sub-groups, 0 group commands

Cloning the Group

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

Subgroups

7.4.8.7.5.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:OBW:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:OBW:CURRent
value: List[float] = driver.multiEval.listPy.spectrum.obw.current.fetch()
```

Return the occupied bandwidth for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per measured segment Range: 0 MHz to 10 MHz, Unit: Hz

7.4.8.7.5.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:OBW:AVERAge
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:OBW:AVERAge
value: List[float] = driver.multiEval.listPy.spectrum.obw.average.fetch()
```

Return the occupied bandwidth for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per measured segment Range: 0 MHz to 10 MHz, Unit: Hz

7.4.8.7.5.3 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:OBW:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:OBW:MAXimum
value: List[float] = driver.multiEval.listPy.spectrum.obw.maximum.fetch()
```

Return the occupied bandwidth for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return obw: float Comma-separated list of values, one per measured segment Range: 0 MHz to 10 MHz, Unit: Hz

7.4.8.7.6 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:CURRent
```

class Current

Current 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Carrier_Power: List[float]: float Power at the nominal carrier frequency in uplink Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_2: List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_1: List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm

- **Aclr_Plus_1**: List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Plus_2**: List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Obw**: List[float]: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- **Emask_Margin_Ab**: List[float]: No parameter help available
- **Emask_Margin_Bc**: List[float]: No parameter help available
- **Emask_Margin_Cd**: List[float]: No parameter help available
- **Emask_Margin_Ef**: List[float]: No parameter help available
- **Emask_Margin_Fe**: List[float]: No parameter help available
- **Emask_Margin_Dc**: List[float]: No parameter help available
- **Emask_Margin_Cb**: List[float]: No parameter help available
- **Emask_Margin_Ba**: List[float]: No parameter help available
- **Ue_Power**: List[float]: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- **Emask_Margin_Had**: List[float]: No parameter help available
- **Emask_Margin_Hda**: List[float]: No parameter help available

fetch(*aclr_mode*: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:CURRent
value: FetchStruct = driver.multiEval.listPy.spectrum.current.fetch(aclr_mode =
enums.AclrMode.Absolute)
```

Returns the ACLR power and spectrum emission single value results in list mode. The values listed below in curly brackets { } are returned for the segments { ... }seg 1, { ... }seg 2, ..., { ... }seg n, with n determined by method RsCmwWcdmaMeas. Configure.MultiEval.ListPy.count.

param aclr_mode Absolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

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

7.4.8.7.7 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:AVERAge
```

class Average

Average 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.
- **Return_Code:** List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- **Carrier_Power:** List[float]: float Power at the nominal carrier frequency in uplink Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Minus_2:** List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Minus_1:** List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Plus_1:** List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Aclr_Plus_2:** List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- **Obw:** List[float]: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- **Emask_Margin_Ab:** List[float]: No parameter help available
- **Emask_Margin_Bc:** List[float]: No parameter help available
- **Emask_Margin_Cd:** List[float]: No parameter help available
- **Emask_Margin_Ef:** List[float]: No parameter help available
- **Emask_Margin_Fe:** List[float]: No parameter help available
- **Emask_Margin_Dc:** List[float]: No parameter help available
- **Emask_Margin_Cb:** List[float]: No parameter help available
- **Emask_Margin_Ba:** List[float]: No parameter help available
- **Ue_Power:** List[float]: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- **Emask_Margin_Had:** List[float]: No parameter help available
- **Emask_Margin_Hda:** List[float]: No parameter help available

fetch(aclr_mode: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:AVERage
value: FetchStruct = driver.multiEval.listPy.spectrum.average.fetch(aclr_mode =
enums.AclrMode.Absolute)
```

Returns the ACLR power and spectrum emission single value results in list mode. The values listed below in curly brackets { } are returned for the segments { ... }seg 1, { ... }seg 2, ..., { ... }seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy.count.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as absolute value RELative: ACLR power displayed in dB relative to carrier power

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

7.4.8.7.8 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:SPECTrum:MAXimum
```

class Maximum

Maximum 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Carrier_Power: List[float]: float Power at the nominal carrier frequency in uplink Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_2: List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Minus_1: List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_1: List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Aclr_Plus_2: List[float]: float Power of the adjacent channels (± 1 st adjacent channels at ± 5 MHz from the UL frequency, ± 2 nd adjacent channels at ± 10 MHz from the UL frequency) Range: -100 dBm to 55 dBm, Unit: dBm
- Obw: List[float]: float Occupied bandwidth Range: 0 MHz to 10 MHz, Unit: Hz
- Emask_Margin_Ab: List[float]: No parameter help available
- Emask_Margin_Bc: List[float]: No parameter help available
- Emask_Margin_Cd: List[float]: No parameter help available
- Emask_Margin_Ef: List[float]: No parameter help available
- Emask_Margin_Fe: List[float]: No parameter help available
- Emask_Margin_Dc: List[float]: No parameter help available
- Emask_Margin_Cb: List[float]: No parameter help available
- Emask_Margin_Ba: List[float]: No parameter help available
- Ue_Power: List[float]: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm
- Emask_Margin_Had: List[float]: No parameter help available
- Emask_Margin_Hda: List[float]: No parameter help available

fetch(*aclr_mode*: Optional[RsCmwWcdmaMeas.enums.AclrMode] = None) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:MAXimum
value: FetchStruct = driver.multiEval.listPy.spectrum.maximum.fetch(aclr_mode =
enums.AcLrMode.ABSolute)
```

Returns the ACLR power and spectrum emission single value results in list mode. The values listed below in curly brackets { } are returned for the segments { ... }seg 1, { ... }seg 2, ..., { ... }seg n, with n determined by method RsCmwWcdmaMeas. Configure.MultiEval.ListPy.count.

param aclr_mode ABSolute | RELative ABSolute: ACLR power displayed in dBm as
absolute value RELative: ACLR power displayed in dB relative to carrier power

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

7.4.8.8 Modulation

class Modulation

Modulation commands group definition. 48 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.modulation.clone()
```

Subgroups

7.4.8.8.1 Evm

class Evm

Evm commands group definition. 8 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

7.4.8.8.1.1 Rms

class Rms

Rms commands group definition. 4 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

7.4.8.8.1.2 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:CURRENT
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.current.fetch()
```

Return error vector magnitude RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per measured segment
Range: 0 % to 100 %, Unit: %

7.4.8.8.1.3 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:AVERage
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.average.fetch()
```

Return error vector magnitude RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per measured segment
Range: 0 % to 100 %, Unit: %

7.4.8.8.1.4 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.maximum.fetch()
```

Return error vector magnitude RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per measured segment
Range: 0 % to 100 %, Unit: %

7.4.8.8.1.5 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:RMS:SDEviation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.evm.rms.standardDev.
↪fetch()
```

Return error vector magnitude RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_rms: float Comma-separated list of values, one per measured segment
Range: 0 % to 100 %, Unit: %

7.4.8.8.1.6 Peak

class Peak

Peak commands group definition. 4 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

7.4.8.8.1.7 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:PEAK:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:CURRENT
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.current.fetch()
```

Return error vector magnitude peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per measured segment
Range: 0 % to 100 %, Unit: %

7.4.8.8.1.8 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:PEAK:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:AVERage
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.average.fetch()
```

Return error vector magnitude peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per measured segment
Range: 0 % to 100 %, Unit: %

7.4.8.8.1.9 Maximum

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:PEAK:MAXimum`

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.maximum.fetch()
```

Return error vector magnitude peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per measured segment
Range: 0 % to 100 %, Unit: %

7.4.8.8.1.10 StandardDev

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:EVM:PEAK:SDEviation`

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.evm.peak.standardDev.
↪fetch()
```

Return error vector magnitude peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_peak: float Comma-separated list of values, one per measured segment
Range: 0 % to 100 %, Unit: %

7.4.8.8.2 Merror

class Merror

Merror commands group definition. 8 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

7.4.8.8.2.1 Rms

class Rms

Rms commands group definition. 4 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

7.4.8.8.2.2 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:CURRent
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.current.
↪fetch()
```

Return magnitude error RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_rms: float Comma-separated list of values, one per measured segment Range: 0 % to 100 %, Unit: %

7.4.8.8.2.3 Average

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODulation:MERRor:RMS:AVERage`

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:MODulation:MERRor:RMS:AVERage
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.average.
↪fetch()
```

Return magnitude error RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_rms: float Comma-separated list of values, one per measured segment Range: 0 % to 100 %, Unit: %

7.4.8.8.2.4 Maximum

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODulation:MERRor:RMS:MAXimum`

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:MODulation:MERRor:RMS:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.maximum.
↪fetch()
```

Return magnitude error RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_rms: float Comma-separated list of values, one per measured segment Range: 0 % to 100 %, Unit: %

7.4.8.8.2.5 StandardDev

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:RMS:SDEViation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.merror.rms.standardDev.
↪fetch()
```

Return magnitude error RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_rms: float Comma-separated list of values, one per measured segment Range: 0 % to 100 %, Unit: %

7.4.8.8.2.6 Peak

class Peak

Peak commands group definition. 4 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

7.4.8.8.2.7 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:MERRor:PEAK:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:MERRor:PEAK:CURRent
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.current.
↪fetch()
```

Return magnitude error peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_peak: float Comma-separated list of values, one per measured segment Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEVIation: 0 % to 50 %) ,
Unit: %

7.4.8.8.2.8 Average

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODulation:MERRor:PEAK:AVERage`

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEValuation:LIST:MODulation:MERRor:PEAK:AVERage
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.average.
↳fetch()
```

Return magnitude error peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_peak: float Comma-separated list of values, one per measured segment Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEVIation: 0 % to 50 %) ,
Unit: %

7.4.8.8.2.9 Maximum

SCPI Commands

`FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODulation:MERRor:PEAK:MAXimum`

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEValuation:LIST:MODulation:MERRor:PEAK:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.maximum.
↳fetch()
```

Return magnitude error peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_peak: float Comma-separated list of values, one per measured segment Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %), Unit: %

7.4.8.8.2.10 StandardDev

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODulation:MERRor:PEAK:SDEViation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEValuation:LIST:MODulation:MERRor:PEAK:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.merror.peak.standardDev.
↳fetch()
```

Return magnitude error peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_peak: float Comma-separated list of values, one per measured segment Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %), Unit: %

7.4.8.8.3 Perror

class Perror

Perror commands group definition. 8 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

7.4.8.8.3.1 Rms

class Rms

Rms commands group definition. 4 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

7.4.8.8.3.2 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:CURRent
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.current.
↪fetch()
```

Return phase error RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_rms: float Comma-separated list of values, one per measured segment Range: 0 deg to 180 deg, Unit: deg

7.4.8.8.3.3 Average

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:AVERAge
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:AVERAge
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.average.
↪fetch()
```

Return phase error RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_rms: float Comma-separated list of values, one per measured segment Range: 0 deg to 180 deg, Unit: deg

7.4.8.8.3.4 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.maximum.
↪fetch()
```

Return phase error RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_rms: float Comma-separated list of values, one per measured segment Range: 0 deg to 180 deg, Unit: deg

7.4.8.8.3.5 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:RMS:SDEviation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:PERRor:RMS:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.perror.rms.standardDev.
↪fetch()
```

Return phase error RMS values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_rms: float Comma-separated list of values, one per measured segment Range: 0 deg to 180 deg, Unit: deg

7.4.8.8.3.6 Peak

class Peak

Peak commands group definition. 4 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

7.4.8.8.3.7 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:PERRor:PEAK:CURRent
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.current.
↪fetch()
```

Return phase error peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_peak: float Comma-separated list of values, one per measured segment Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg

7.4.8.8.3.8 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:PERRor:PEAK:AVERage
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.average.
↳fetch()
```

Return phase error peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_peak: float Comma-separated list of values, one per measured segment Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg

7.4.8.8.3.9 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:PERRor:PEAK:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.maximum.
↳fetch()
```

Return phase error peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_peak: float Comma-separated list of values, one per measured segment Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg

7.4.8.8.3.10 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:SDEViation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:PERRor:PEAK:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.perror.peak.standardDev.
↳fetch()
```

Return phase error peak values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_peak: float Comma-separated list of values, one per measured segment Range: -180 deg to 180 deg (AVERage: 0 deg to 180 deg, SDEViation: 0 deg to 90 deg) , Unit: deg

7.4.8.8.4 IqOffset

class IqOffset

IqOffset commands group definition. 4 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

7.4.8.8.4.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQOffset:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:IQOffset:CURRENT
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.current.fetch()
```

Return I/Q origin offset values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.8.4.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:IQOffset:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEValuation:LIST:MODulation:IQOffset:AVERage
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.average.fetch()
```

Return I/Q origin offset values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.8.4.3 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEValuation:LIST:MODulation:IQOffset:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEValuation:LIST:MODulation:IQOffset:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.maximum.fetch()
```

Return I/Q origin offset values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.8.4 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQOffset:SDEviation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:IQOffset:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.iqOffset.standardDev.
↪fetch()
```

Return I/Q origin offset values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return iq_offset: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.8.5 IqImbalance

class IqImbalance

IqImbalance commands group definition. 4 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

7.4.8.8.5.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:CURRENT
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:CURRENT
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.current.
↪fetch()
```

Return I/Q imbalance values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.8.5.2 Average

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODulation:IQIMbalance:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:MODulation:IQIMbalance:AVERage
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.average.
↪fetch()
```

Return I/Q imbalance values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.8.5.3 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODulation:IQIMbalance:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:MODulation:IQIMbalance:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.maximum.
↪fetch()
```

Return I/Q imbalance values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.8.5.4 StandardDev

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:IQIMbalance:SDEviation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.iqImbalance.standardDev.
↪fetch()
```

Return I/Q imbalance values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return iq_imbalance: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.8.6 FreqError

class FreqError

FreqError commands group definition. 4 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

7.4.8.8.6.1 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:FERRor:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:FERRor:CURRent
value: List[float] = driver.multiEval.listPy.modulation.freqError.current.
↪fetch()
```

Return carrier frequency error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return carrier_freq_err: float Comma-separated list of values, one per measured segment Range: -60000 Hz to 60000 Hz, Unit: Hz

7.4.8.8.6.2 Average

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:FERRor:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:FERRor:AVERage
value: List[float] = driver.multiEval.listPy.modulation.freqError.average.
↪fetch()
```

Return carrier frequency error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return carrier_freq_err: float Comma-separated list of values, one per measured segment Range: -60000 Hz to 60000 Hz, Unit: Hz

7.4.8.8.6.3 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:FERRor:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:FERRor:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.freqError.maximum.
↪fetch()
```

Return carrier frequency error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return carrier_freq_err: float Comma-separated list of values, one per measured segment Range: -60000 Hz to 60000 Hz, Unit: Hz

7.4.8.8.6.4 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:FERRor:SDEViation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:FERRor:SDEViation
value: List[float] = driver.multiEval.listPy.modulation.freqError.standardDev.
↪fetch()
```

Return carrier frequency error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return carrier_freq_err: float Comma-separated list of values, one per measured segment Range: -60000 Hz to 60000 Hz, Unit: Hz

7.4.8.8.7 TtError

class TtError

TtError commands group definition. 4 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.ttError.clone()
```

Subgroups

7.4.8.8.7.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TTERror:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:TTERror:CURRent
value: List[float] = driver.multiEval.listPy.modulation.ttError.current.fetch()
```

Return transmit time error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return transmit_time_err: float Comma-separated list of values, one per measured segment Range: -250 chips to 250 chips, Unit: chip

7.4.8.8.7.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TTError:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:TTError:AVERage
value: List[float] = driver.multiEval.listPy.modulation.ttError.average.fetch()
```

Return transmit time error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return transmit_time_err: float Comma-separated list of values, one per measured segment Range: -250 chips to 250 chips, Unit: chip

7.4.8.8.7.3 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TTError:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:TTError:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.ttError.maximum.fetch()
```

Return transmit time error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return transmit_time_err: float Comma-separated list of values, one per measured segment Range: -250 chips to 250 chips, Unit: chip

7.4.8.8.7.4 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:TTError:SDEviation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:TTError:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.ttError.standardDev.
↪fetch()
```

Return transmit time error values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return transmit_time_err: float Comma-separated list of values, one per measured segment Range: -250 chips to 250 chips, Unit: chip

7.4.8.8.8 UePower

class UePower

UePower commands group definition. 4 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.uePower.clone()
```

Subgroups

7.4.8.8.8.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:UEPower:CURRent
```

class Current

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:UEPower:CURRent
value: List[float] = driver.multiEval.listPy.modulation.uePower.current.fetch()
```


Return user equipment power values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.8.2 Average

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:UEPower:AVERage
```

class Average

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:UEPower:AVERage
value: List[float] = driver.multiEval.listPy.modulation.uePower.average.fetch()
```

Return user equipment power values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.8.3 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:UEPower:MAXimum
```

class Maximum

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:UEPower:MAXimum
value: List[float] = driver.multiEval.listPy.modulation.uePower.maximum.fetch()
```

Return user equipment power values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.8.4 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:UEPower:SDEviation
```

class StandardDev

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

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
→:MEvaluation:LIST:MODulation:UEPower:SDEviation
value: List[float] = driver.multiEval.listPy.modulation.uePower.standardDev.
→fetch()
```

Return user equipment power values for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float Comma-separated list of values, one per measured segment
Range: -100 dBm to 55 dBm, Unit: dBm

7.4.8.8.9 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:MODulation:CURRENT
```

class Current

Current 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- 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: %
- Mag_Error_Rms: List[float]: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: List[float]: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEviation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: List[float]: No parameter help available
- Phase_Error_Peak: List[float]: No parameter help available
- 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
- Carrier_Freq_Err: List[float]: float Carrier frequency error Range: -60000 Hz to 60000 Hz, Unit: Hz
- Transmit_Time_Err: List[float]: No parameter help available
- Ue_Power: List[float]: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:MODULATION:CURRENT
value: FetchStruct = driver.multiEval.listPy.modulation.current.fetch()
```

Return modulation single value results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

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

7.4.8.8.10 Average

SCPI Commands

```
FETCH:WCDMA:MEASUREMENT<Instance>:MEVALUATION:LIST:MODULATION:AVERAGE
```

class Average

Average 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- 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: %
- Mag_Error_Rms: List[float]: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: List[float]: float Magnitude error peak value Range: -100 % to 100 % (AVERAGE: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: List[float]: No parameter help available
- Phase_Error_Peak: List[float]: No parameter help available
- 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
- Carrier_Freq_Err: List[float]: float Carrier frequency error Range: -60000 Hz to 60000 Hz, Unit: Hz
- Transmit_Time_Err: List[float]: No parameter help available

- Ue_Power: List[float]: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm

fetch() → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODulation:AVERage
value: FetchStruct = driver.multiEval.listPy.modulation.average.fetch()
```

Return modulation single value results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

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

7.4.8.8.11 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEvaluation:LIST:MODulation:MAXimum
```

class Maximum

Maximum 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- 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: %
- Mag_Error_Rms: List[float]: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: List[float]: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: List[float]: No parameter help available
- Phase_Error_Peak: List[float]: No parameter help available
- 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
- Carrier_Freq_Err: List[float]: float Carrier frequency error Range: -60000 Hz to 60000 Hz, Unit: Hz
- Transmit_Time_Err: List[float]: No parameter help available
- Ue_Power: List[float]: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm

fetch() → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:MAXimum
value: FetchStruct = driver.multiEval.listPy.modulation.maximum.fetch()
```

Return modulation single value results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

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

7.4.8.8.12 StandardDev

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODulation:SDEViation
```

class StandardDev

StandardDev 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- 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: %
- Mag_Error_Rms: List[float]: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: List[float]: float Magnitude error peak value Range: -100 % to 100 % (AVERage: 0% to 100 %, SDEViation: 0 % to 50 %) , Unit: %
- Phase_Error_Rms: List[float]: No parameter help available
- Phase_Error_Peak: List[float]: No parameter help available
- 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
- Carrier_Freq_Err: List[float]: float Carrier frequency error Range: -60000 Hz to 60000 Hz, Unit: Hz
- Transmit_Time_Err: List[float]: No parameter help available
- Ue_Power: List[float]: float User equipment power Range: -100 dBm to 55 dBm, Unit: dBm

fetch() → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:SDEViation
value: FetchStruct = driver.multiEval.listPy.modulation.standardDev.fetch()
```

Return modulation single value results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.8.9 CdError

class CdError

CdError commands group definition. 24 total commands, 9 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cdError.clone()
```

Subgroups

7.4.8.9.1 Dpcch

class Dpcch

Dpcch commands group definition. 4 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.cdError.dpcch.clone()
```

Subgroups

7.4.8.9.1.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:DPCCh:CURRENT
```

class Current

Current commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:CURRENT
value: List[float] = driver.multiEval.listPy.cdError.dpcch.current.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpcch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.1.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:DPCCh:AVERage
```

class Average

Average commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:AVERage
value: List[float] = driver.multiEval.listPy.cdError.dpcch.average.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpcch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.1.3 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:DPCCh:MAXimum
```

class Maximum

Maximum commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:MAXimum
value: List[float] = driver.multiEval.listPy.cdError.dpcch.maximum.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpcch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.1.4 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:DPCh:SDEviation
```

class StandardDev

StandardDev commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDError:DPCh:SDEviation
value: List[float] = driver.multiEval.listPy.cdError.dpcch.standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.9.2 Dpdch

class Dpdch

Dpdch commands group definition. 4 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.cdError.dpdch.clone()
```

Subgroups

7.4.8.9.2.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:DPDCh:CURRent
```

class Current

Current commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:CURRent
value: List[float] = driver.multiEval.listPy.cdError.dpdch.current.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpdch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.9.2.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:DPDCh:AVERage
```

class Average

Average commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:AVERage
value: List[float] = driver.multiEval.listPy.cdError.dpdch.average.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpdch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.9.2.3 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:DPDCh:MAXimum
```

class Maximum

Maximum commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:MAXimum
value: List[float] = driver.multiEval.listPy.cdError.dpdch.maximum.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpdch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.9.2.4 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:DPDCh:SDEviation
```

class StandardDev

StandardDev commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDError:DPDCh:SDEviation
value: List[float] = driver.multiEval.listPy.cdError.dpdch.standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return dpdch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.9.3 Hsdpcch

class Hsdpcch

Hsdpcch commands group definition. 4 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.cdError.hsdpcch.clone()
```

Subgroups

7.4.8.9.3.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:HSDPcch:CURRENT
```

class Current

Current commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDError:HSDPcch:CURRENT
value: List[float] = driver.multiEval.listPy.cdError.hsdpcch.current.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return hsdpcch: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.3.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDERror:HSDPcch:AVERage
```

class Average

Average commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDERror:HSDPcch:AVERage
value: List[float] = driver.multiEval.listPy.cdError.hsdpcch.average.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return hsdpcch: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.3.3 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDERror:HSDPcch:MAXimum
```

class Maximum

Maximum commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDERror:HSDPcch:MAXimum
value: List[float] = driver.multiEval.listPy.cdError.hsdpcch.maximum.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return hsdpcch: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.3.4 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:HSDPcch:SDEviation
```

class StandardDev

StandardDev commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDError:HSDPcch:SDEviation
value: List[float] = driver.multiEval.listPy.cdError.hsdpcch.standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return hsdpcch: float Comma-separated list of values, one per measured segment
Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.4 Edpcch

class Edpcch

Edpcch commands group definition. 4 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.cdError.edpcch.clone()
```

Subgroups

7.4.8.9.4.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:EDPCch:CURRENT
```

class Current

Current commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDError:EDPCch:CURRENT
value: List[float] = driver.multiEval.listPy.cdError.edpcch.current.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return edpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.9.4.2 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDERror:EDPCch:AVERage
```

class Average

Average commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDERror:EDPCch:AVERage
value: List[float] = driver.multiEval.listPy.cdError.edpcch.average.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return edpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.9.4.3 Maximum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDERror:EDPCch:MAXimum
```

class Maximum

Maximum commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:CDERror:EDPCch:MAXimum
value: List[float] = driver.multiEval.listPy.cdError.edpcch.maximum.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return edpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.9.4 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:EDPCch:SDEviation
```

class StandardDev

StandardDev commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
→:MEvaluation:LIST:CDError:EDPCch:SDEviation
value: List[float] = driver.multiEval.listPy.cdError.edpcch.standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return edpcch: float Comma-separated list of values, one per measured segment Range:
-100 dB to 0 dB, Unit: dB

7.4.8.9.5 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.multiEval.listPy.cdError.edpdch.repcap_edpdChannel_get()
driver.multiEval.listPy.cdError.edpdch.repcap_edpdChannel_set(repcap.EdpdChannel.Nr1)
```

class Edpdch

Edpdch commands group definition. 4 total commands, 4 Sub-groups, 0 group commands Repeated Capability:
EdpdChannel, default value after init: EdpdChannel.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.multiEval.listPy.cdError.edpdch.clone()
```

Subgroups

7.4.8.9.5.1 Current

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:EDPDch<EdpdChannel>:CURRENT
```

class Current

Current commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch(edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDError:EDPDch<nr>
↪:CURRent
value: List[float] = driver.multiEval.listPy.cdError.edpdch.current.
↪fetch(edpdChannel = repcap.EdpdChannel.Default)
```

Return RMS CDP and CDE vs. slot values for a selected E-DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.5.2 Average

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDError:EDPDch<EdpdChannel>:AVERage
```

class Average

Average commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch(edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDError:EDPDch<nr>
↪:AVERage
value: List[float] = driver.multiEval.listPy.cdError.edpdch.average.
↪fetch(edpdChannel = repcap.EdpdChannel.Default)
```

Return RMS CDP and CDE vs. slot values for a selected E-DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.5.3 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDError:EDPDch<EdpdChannel>:MAXimum
```

class Maximum

Maximum commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch(edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDError:EDPDch<nr>
↳:MAXimum
value: List[float] = driver.multiEval.listPy.cdError.edpdch.maximum.
↳fetch(edpdChannel = repcap.EdpdChannel.Default)
```

Return RMS CDP and CDE vs. slot values for a selected E-DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.5.4 StandardDev

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDError:EDPDch<EdpdChannel>:SDEviation
```

class StandardDev

StandardDev commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

fetch(edpdChannel=<EdpdChannel.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDError:EDPDch<nr>
↳:SDEviation
value: List[float] = driver.multiEval.listPy.cdError.edpdch.standardDev.
↳fetch(edpdChannel = repcap.EdpdChannel.Default)
```

Return RMS CDP and CDE vs. slot values for a selected E-DPDCH for all measured list mode segments.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param edpdChannel optional repeated capability selector. Default value: Nr1 (settable in the interface 'Edpdch')

return edpdch: float Comma-separated list of values, one per measured segment Range: -100 dB to 0 dB, Unit: dB

7.4.8.9.6 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDError:CURRent
```

class Current

Current 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.
- **Return_Code:** List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- **Dpcch:** List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Dpdch:** List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Hsdpcch:** List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpcch:** List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpdch_1:** List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpdch_2:** List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpdch_3:** List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpdch_4:** List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:CURRENT
value: FetchStruct = driver.multiEval.listPy.cdError.current.fetch()
```

Return the RMS CDE vs. slot results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.8.9.7 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:AVERage
```

class Average

Average 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.

- **Return_Code**: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- **Dpcch**: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Dpdch**: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Hsdpcch**: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpcch**: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpdch_1**: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpdch_2**: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpdch_3**: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- **Edpdch_4**: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch() → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDError:AVERage
value: FetchStruct = driver.multiEval.listPy.cdError.average.fetch()
```

Return the RMS CDE vs. slot results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.8.9.8 Maximum

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:MEValuation:LIST:CDError:MAXimum
```

class Maximum

Maximum 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.
- **Return_Code**: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- **Dpcch**: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Dpdch: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpcch: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:MAXimum
value: FetchStruct = driver.multiEval.listPy.cdError.maximum.fetch()
```

Return the RMS CDE vs. slot results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.4.8.9.9 StandardDev

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:MEvaluation:LIST:CDError:SDEViation
```

class StandardDev

StandardDev 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.
- Return_Code: List[int]: decimal Reliability indicator for the segment. The meaning of the returned values is the same as for the common reliability indicator, see previous parameter.
- Dpcch: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Dpdch: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Hsdpcch: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

- Edpcch: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_1: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_2: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_3: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB
- Edpdch_4: List[float]: float RMS CDE values for the indicated channels Range: -100 dB to 0 dB (SDEViation 0 dB to 50 dB) , Unit: dB

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:SDEViation
value: FetchStruct = driver.multiEval.listPy.cdError.standardDev.fetch()
```

Return the RMS CDE vs. slot results in list mode. The values listed below in curly brackets {} are returned for the segments {...}seg 1, {...}seg 2, ..., {...}seg n, with n determined by method RsCmwWcdmaMeas.Configure.MultiEval.ListPy. count.

return structure: for return value, see the help for FetchStruct structure arguments.

7.5 Tpc

SCPI Commands

```
STOP:WCDMA:MEASurement<Instance>:TPC
ABORT:WCDMA:MEASurement<Instance>:TPC
INITiate:WCDMA:MEASurement<Instance>:TPC
```

class Tpc

Tpc commands group definition. 54 total commands, 4 Sub-groups, 3 group commands

abort() → None

```
# SCPI: ABORT:WCDMA:MEASurement<instance>:TPC
driver.tpc.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:WCDMa:MEASurement<instance>:TPC
driver.tpc.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 RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

initiate() → None

```
# SCPI: INITiate:WCDMa:MEASurement<instance>:TPC
driver.tpc.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:WCDMa:MEASurement<instance>:TPC
driver.tpc.initiate_with_opc()
```

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.

Same as initiate, but waits for the operation to complete before continuing further. Use the RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

stop() → None

```

# SCPI: STOP:WCDMA:MEASurement<instance>:TPC
driver.tpc.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.
- 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:WCDMA:MEASurement<instance>:TPC
driver.tpc.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 RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.tpc.clone()
```

Subgroups

7.5.1 Dhib

class Dhib

Dhib 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.tpc.dhib.clone()
```

Subgroups

7.5.1.1 Maximum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:TPC:DHIB:MAXimum
FETCh:WCDMa:MEASurement<Instance>:TPC:DHIB:MAXimum
CALCulate:WCDMa:MEASurement<Instance>:TPC:DHIB: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'
- Carrier_Ch_Power: float: float Level of the uplink carrier, where the UE transmits at the maximal output power Range: -100 dBm to 40 dBm, Unit: dBm
- Inband_Emission: float: float Relative level of the other uplink carrier transmitting at minimal output power Range: -99 dB to 99 dB, Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Carrier_Ch_Power: float: float Level of the uplink carrier, where the UE transmits at the maximal output power Range: -100 dBm to 40 dBm, Unit: dBm
- Inband_Emission: float: float Relative level of the other uplink carrier transmitting at minimal output power Range: -99 dB to 99 dB, Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:DHIB:MAXimum
value: CalculateStruct = driver.tpc.dhib.maximum.calculate()
```

Return the dual carrier in-band emission results. The minimum, maximum and average results 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:WCDMa:MEASurement<instance>:TPC:DHIB:MAXimum
value: ResultData = driver.tpc.dhib.maximum.fetch()
```

Return the dual carrier in-band emission results. The minimum, maximum and average results 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:WCDMa:MEASurement<instance>:TPC:DHIB:MAXimum
value: ResultData = driver.tpc.dhib.maximum.read()
```

Return the dual carrier in-band emission results. The minimum, maximum and average results 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.

7.5.1.2 Minimum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:TPC:DHIB:MINimum
FETCh:WCDMa:MEASurement<Instance>:TPC:DHIB:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Carrier_Ch_Power: float: float Level of the uplink carrier, where the UE transmits at the maximal output power Range: -100 dBm to 40 dBm, Unit: dBm
- Inband_Emission: float: float Relative level of the other uplink carrier transmitting at minimal output power Range: -99 dB to 99 dB, Unit: dB

fetch() → ResultData


```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:DHIB:MINimum
value: ResultData = driver.tpc.dhib.minimum.fetch()
```

Return the dual carrier in-band emission results. The minimum, maximum and average results 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:WCDMa:MEASurement<instance>:TPC:DHIB:MINimum
value: ResultData = driver.tpc.dhib.minimum.read()
```

Return the dual carrier in-band emission results. The minimum, maximum and average results 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.

7.5.1.3 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:TPC:DHIB:AVERage
FETCh:WCDMa:MEASurement<Instance>:TPC:DHIB:AVERage
CALCulate:WCDMa:MEASurement<Instance>:TPC:DHIB: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'
- Carrier_Ch_Power: float: float Level of the uplink carrier, where the UE transmits at the maximal output power Range: -100 dBm to 40 dBm, Unit: dBm
- Inband_Emission: float: float Relative level of the other uplink carrier transmitting at minimal output power Range: -99 dB to 99 dB, Unit: dB

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Carrier_Ch_Power: float: float Level of the uplink carrier, where the UE transmits at the maximal output power Range: -100 dBm to 40 dBm, Unit: dBm
- Inband_Emission: float: float Relative level of the other uplink carrier transmitting at minimal output power Range: -99 dB to 99 dB, Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:DHIB:AVERage
value: CalculateStruct = driver.tpc.dhib.average.calculate()
```

Return the dual carrier in-band emission results. The minimum, maximum and average results 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:WCDMa:MEASurement<instance>:TPC:DHIB:AVERage
value: ResultData = driver.tpc.dhib.average.fetch()
```

Return the dual carrier in-band emission results. The minimum, maximum and average results 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:WCDMa:MEASurement<instance>:TPC:DHIB:AVERage
value: ResultData = driver.tpc.dhib.average.read()
```

Return the dual carrier in-band emission results. The minimum, maximum and average results 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.

7.5.1.4 Statistics

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:TPC:DHIB:STATistics
FETCh:WCDMa:MEASurement<Instance>:TPC:DHIB:STATistics
```

class Statistics

Statistics commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → float

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:DHIB:STATistics
value: float = driver.tpc.dhib.statistics.fetch()
```

Return the ‘Statistics’ values, indicating how many trace values have been considered to derive the maximum, minimum and average dual carrier in-band emission results. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return statistics: float Range: 0 to 1000

read() → float

```
# SCPI: READ:WCDMA:MEASurement<instance>:TPC:DHIB:STATistics
value: float = driver.tpc.dhib.statistics.read()
```

Return the ‘Statistics’ values, indicating how many trace values have been considered to derive the maximum, minimum and average dual carrier in-band emission results. The number to the left of each result parameter is provided for easy identification of the parameter position within the result array.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return statistics: float Range: 0 to 1000

7.5.1.5 Minimumc

SCPI Commands

```
CALCulate:WCDMA:MEASurement<Instance>:TPC:DHIB:MINimumc
```

class Minimumc

Minimumc commands group definition. 1 total commands, 0 Sub-groups, 1 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Carrier_Ch_Power: float: float Level of the uplink carrier, where the UE transmits at the maximal output power Range: -100 dBm to 40 dBm, Unit: dBm
- Inband_Emission: float: float Relative level of the other uplink carrier transmitting at minimal output power Range: -99 dB to 99 dB, Unit: dB

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMA:MEASurement<instance>:TPC:DHIB:MINimumc
value: CalculateStruct = driver.tpc.dhib.minimumc.calculate()
```

Return the dual carrier in-band emission results. The minimum, maximum and average results 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.

7.5.2 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.tpc.carrier.repcap_carrier_get()
driver.tpc.carrier.repcap_carrier_set(repcap.Carrier.Nr1)
```

class Carrier

Carrier commands group definition. 28 total commands, 3 Sub-groups, 0 group commands Repeated Capability:
Carrier, default value after init: Carrier.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.tpc.carrier.clone()
```

Subgroups**7.5.2.1 Psteps****class Psteps**

Psteps commands group definition. 11 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.tpc.carrier.psteps.clone()
```

Subgroups**7.5.2.1.1 Maximum****SCPI Commands**

```
READ:WCDMA:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps:MAXimum
FETCh:WCDMA:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps:MAXimum
CALCulate:WCDMA:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps: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'
- Pwr_Steps_0_Db: float: No parameter help available
- Pwr_Steps_B_1_Db: float: No parameter help available
- Pwr_Steps_Cm_1_Db: float: No parameter help available
- Pwr_Steps_Group_A: float: No parameter help available
- Pwr_Steps_Group_B: float: No parameter help available
- Pwr_Steps_Group_C: float: No parameter help available
- Start_Slot_Group_A: float: No parameter help available
- Pwr_Steps_Eg: float: No parameter help available

- Pwr_Steps_Fh: float: No parameter help available
- Pwr_Steps_Group_Eg: float: No parameter help available
- Pwr_Steps_Group_Fh: float: No parameter help available
- Start_Slot_Group_Eg: float: No parameter help available
- Start_Slot_Group_Fh: float: No parameter help available
- Pwr_Steps_Up: float: No parameter help available
- Pwr_Steps_Down: float: No parameter help available
- Epwr_Steps_B_1_D_B: enums.ResultStatus2: No parameter help available
- Epwr_Steps_Cm_1_D_B: enums.ResultStatus2: No parameter help available
- Epwr_Steps_Eg: enums.ResultStatus2: No parameter help available
- Epwr_Steps_Fh: enums.ResultStatus2: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pwr_Steps_0_Db: float: No parameter help available
- Pwr_Steps_B_1_Db: float: No parameter help available
- Pwr_Steps_Cm_1_Db: float: No parameter help available
- Pwr_Steps_Group_A: float: No parameter help available
- Pwr_Steps_Group_B: float: No parameter help available
- Pwr_Steps_Group_C: float: No parameter help available
- Start_Slot_Group_A: int: No parameter help available
- Pwr_Steps_Eg: float: No parameter help available
- Pwr_Steps_Fh: float: No parameter help available
- Pwr_Steps_Group_Eg: float: No parameter help available
- Pwr_Steps_Group_Fh: float: No parameter help available
- Start_Slot_Group_Eg: int: No parameter help available
- Start_Slot_Group_Fh: int: No parameter help available
- Pwr_Steps_Up: float: No parameter help available
- Pwr_Steps_Down: float: No parameter help available
- Init_Pwr_Step: float: No parameter help available
- Rpwr_Steps: float: No parameter help available
- Rpwr_Steps_Group: float: No parameter help available
- Pwr_Step_Ncm_Cm: float: No parameter help available
- Pwr_Step_Cm_Ncm: float: No parameter help available
- Epwr_Steps_B_1_D_B: float: No parameter help available
- Epwr_Steps_Cm_1_D_B: float: No parameter help available

- Epwr_Steps_Eg: float: No parameter help available
- Epwr_Steps_Fh: float: No parameter help available

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:PSTeps:MAXimum
value: CalculateStruct = driver.tpc.carrier.psteps.maximum.calculate(carrier =
↳repcap.Carrier.Default)
```

Return the power step and power step group single value results per carrier. The minimum, maximum and average results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <14_StartFH> and <22_EPStepsB1dB> to <25_EPStepsFH>). 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:MAXimum
value: ResultData = driver.tpc.carrier.psteps.maximum.fetch(carrier = repcap.
↳Carrier.Default)
```

Return the power step and power step group single value results per carrier. The minimum, maximum and average results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <14_StartFH> and <22_EPStepsB1dB> to <25_EPStepsFH>). 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:MAXimum
value: ResultData = driver.tpc.carrier.psteps.maximum.read(carrier = repcap.
↳Carrier.Default)
```

Return the power step and power step group single value results per carrier. The minimum, maximum and average results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <14_StartFH> and <22_EPStepsB1dB> to <25_EPStepsFH>). 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

7.5.2.1.2 Minimum

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps:MINimum
FETCh:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps:MINimum
CALCulate:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps: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’
- Pwr_Steps_0_Db: float: No parameter help available
- Pwr_Steps_B_1_Db: float: No parameter help available
- Pwr_Steps_Cm_1_Db: float: No parameter help available
- Pwr_Steps_Group_A: float: No parameter help available
- Pwr_Steps_Group_B: float: No parameter help available
- Pwr_Steps_Group_C: float: No parameter help available
- Start_Slot_Group_A: float: No parameter help available
- Pwr_Steps_Eg: float: No parameter help available
- Pwr_Steps_Fh: float: No parameter help available
- Pwr_Steps_Group_Eg: float: No parameter help available
- Pwr_Steps_Group_Fh: float: No parameter help available
- Start_Slot_Group_Eg: float: No parameter help available
- Start_Slot_Group_Fh: float: No parameter help available
- Pwr_Steps_Up: float: No parameter help available
- Pwr_Steps_Down: float: No parameter help available
- Epwr_Steps_B_1_D_B: enums.ResultStatus2: No parameter help available
- Epwr_Steps_Cm_1_D_B: enums.ResultStatus2: No parameter help available
- Epwr_Steps_Eg: enums.ResultStatus2: No parameter help available
- Epwr_Steps_Fh: enums.ResultStatus2: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Pwr_Steps_0_Db: float: No parameter help available

- Pwr_Steps_B_1_Db: float: No parameter help available
- Pwr_Steps_Cm_1_Db: float: No parameter help available
- Pwr_Steps_Group_A: float: No parameter help available
- Pwr_Steps_Group_B: float: No parameter help available
- Pwr_Steps_Group_C: float: No parameter help available
- Start_Slot_Group_A: int: No parameter help available
- Pwr_Steps_Eg: float: No parameter help available
- Pwr_Steps_Fh: float: No parameter help available
- Pwr_Steps_Group_Eg: float: No parameter help available
- Pwr_Steps_Group_Fh: float: No parameter help available
- Start_Slot_Group_Eg: int: No parameter help available
- Start_Slot_Group_Fh: int: No parameter help available
- Pwr_Steps_Up: float: No parameter help available
- Pwr_Steps_Down: float: No parameter help available
- Init_Pwr_Step: float: No parameter help available
- Rpwr_Steps: float: No parameter help available
- Rpwr_Steps_Group: float: No parameter help available
- Pwr_Step_Ncm_Cm: float: No parameter help available
- Pwr_Step_Cm_Ncm: float: No parameter help available
- Epwr_Steps_B_1_D_B: float: No parameter help available
- Epwr_Steps_Cm_1_D_B: float: No parameter help available
- Epwr_Steps_Eg: float: No parameter help available
- Epwr_Steps_Fh: float: No parameter help available

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDMA:MEASurement<instance>:TPC:CARRIER<carrier>
↪:PSTeps:MINimum
value: CalculateStruct = driver.tpc.carrier.psteps.minimum.calculate(carrier = ↪
↪repcap.Carrier.Default)
```

Return the power step and power step group single value results per carrier. The minimum, maximum and average results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <14_StartFH> and <22_EPStepsB1dB> to <25_EPStepsFH>) . The values described below are returned by FETCh and READ commands. CAL- Culate commands return limit check results instead, one value for each result listed below.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:MINimum
value: ResultData = driver.tpc.carrier.psteps.minimum.fetch(carrier = repcap.
↳Carrier.Default)
```

Return the power step and power step group single value results per carrier. The minimum, maximum and average results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <14_StartFH> and <22_EPStepsB1dB> to <25_EPStepsFH>). 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:MINimum
value: ResultData = driver.tpc.carrier.psteps.minimum.read(carrier = repcap.
↳Carrier.Default)
```

Return the power step and power step group single value results per carrier. The minimum, maximum and average results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <14_StartFH> and <22_EPStepsB1dB> to <25_EPStepsFH>). 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

7.5.2.1.3 Average

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps:AVERage
FETCh:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps:AVERage
CALCulate:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps: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’
- Pwr_Steps_0_Db: float: No parameter help available

- Pwr_Steps_B_1_Db: float: No parameter help available
- Pwr_Steps_Cm_1_Db: float: No parameter help available
- Pwr_Steps_Group_A: float: No parameter help available
- Pwr_Steps_Group_B: float: No parameter help available
- Pwr_Steps_Group_C: float: No parameter help available
- Start_Slot_Group_A: float: No parameter help available
- Pwr_Steps_Eg: float: No parameter help available
- Pwr_Steps_Fh: float: No parameter help available
- Pwr_Steps_Group_Eg: float: No parameter help available
- Pwr_Steps_Group_Fh: float: No parameter help available
- Start_Slot_Group_Eg: float: No parameter help available
- Start_Slot_Group_Fh: float: No parameter help available
- Pwr_Steps_Up: float: No parameter help available
- Pwr_Steps_Down: float: No parameter help available
- Epwr_Steps_B_1_D_B: enums.ResultStatus2: No parameter help available
- Epwr_Steps_Cm_1_D_B: enums.ResultStatus2: No parameter help available
- Epwr_Steps_Eg: enums.ResultStatus2: No parameter help available
- Epwr_Steps_Fh: enums.ResultStatus2: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Pwr_Steps_0_Db: float: No parameter help available
- Pwr_Steps_B_1_Db: float: No parameter help available
- Pwr_Steps_Cm_1_Db: float: No parameter help available
- Pwr_Steps_Group_A: float: No parameter help available
- Pwr_Steps_Group_B: float: No parameter help available
- Pwr_Steps_Group_C: float: No parameter help available
- Start_Slot_Group_A: int: No parameter help available
- Pwr_Steps_Eg: float: No parameter help available
- Pwr_Steps_Fh: float: No parameter help available
- Pwr_Steps_Group_Eg: float: No parameter help available
- Pwr_Steps_Group_Fh: float: No parameter help available
- Start_Slot_Group_Eg: int: No parameter help available
- Start_Slot_Group_Fh: int: No parameter help available
- Pwr_Steps_Up: float: No parameter help available
- Pwr_Steps_Down: float: No parameter help available

- Init_Pwr_Step: float: No parameter help available
- Rpwr_Steps: float: No parameter help available
- Rpwr_Steps_Group: float: No parameter help available
- Pwr_Step_Ncm_Cm: float: No parameter help available
- Pwr_Step_Cm_Ncm: float: No parameter help available
- Epwr_Steps_B_1_D_B: float: No parameter help available
- Epwr_Steps_Cm_1_D_B: float: No parameter help available
- Epwr_Steps_Eg: float: No parameter help available
- Epwr_Steps_Fh: float: No parameter help available

calculate(carrier=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:TPC:CARRier<carrier>
↳:PSTeps:AVERage
value: CalculateStruct = driver.tpc.carrier.psteps.average.calculate(carrier =
↳repcap.Carrier.Default)
```

Return the power step and power step group single value results per carrier. The minimum, maximum and average results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <14_StartFH> and <22_EPStepsB1dB> to <25_EPStepsFH>). 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDma:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:AVERage
value: ResultData = driver.tpc.carrier.psteps.average.fetch(carrier = repcap.
↳Carrier.Default)
```

Return the power step and power step group single value results per carrier. The minimum, maximum and average results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <14_StartFH> and <22_EPStepsB1dB> to <25_EPStepsFH>). 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

read(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDma:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:AVERage
value: ResultData = driver.tpc.carrier.psteps.average.read(carrier = repcap.
↳Carrier.Default)
```

Return the power step and power step group single value results per carrier. The minimum, maximum and average results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <14_StartFH> and <22_EPStepsB1dB> to <25_EPStepsFH>). 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

7.5.2.1.4 Statistics

SCPI Commands

```
READ:WCDma:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps:STATistics
FETCh:WCDma:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTeps:STATistics
```

class Statistics

Statistics commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Pwr_Steps_0_Db: int: No parameter help available
- Pwr_Steps_B_1_Db: int: No parameter help available
- Pwr_Steps_Cm_1_Db: int: No parameter help available
- Pwr_Steps_Group_A: int: No parameter help available
- Pwr_Steps_Eg: int: No parameter help available
- Pwr_Steps_Fh: int: No parameter help available
- Pwr_Steps_Group_Eg: int: No parameter help available
- Pwr_Steps_Group_Fh: int: No parameter help available
- Pwr_Steps_Up: int: float Power steps up result of ‘Change of TFC’ mode Range: 0 to 5
- Pwr_Steps_Down: int: float Power steps down result of ‘Change of TFC’ mode Range: 0 to 5
- Rpwr_Steps: int: decimal Recovery power steps result of ‘UL Compressed Mode’ - pattern A
- Epwr_Steps_B_1_D_B: int: No parameter help available
- Epwr_Steps_Cm_1_D_B: int: No parameter help available
- Epwr_Steps_Eg: int: No parameter help available
- Epwr_Steps_Fh: int: No parameter help available

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:STATistics
value: ResultData = driver.tpc.carrier.psteps.statistics.fetch(carrier = reprecap.
↳Carrier.Default)
```

Return the ‘Statistics’ values per carrier, indicating how many trace values have been considered to derive the maximum, minimum and average power step and power step group results. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters result values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <9_GroupFH> and <13_EPStepsB1dB> to <16_EPStepsFH>).

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:STATistics
value: ResultData = driver.tpc.carrier.psteps.statistics.read(carrier = reprecap.
↳Carrier.Default)
```

Return the ‘Statistics’ values per carrier, indicating how many trace values have been considered to derive the maximum, minimum and average power step and power step group results. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters result values are available. For the other parameters, only an indicator is returned (e.g. NAV) . ‘Step A’ to ‘step H’ refer to the test steps of the ‘Inner Loop Power Control’ mode (results <2_Step0dB_ABC> to <9_GroupFH> and <13_EPStepsB1dB> to <16_EPStepsFH>).

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

7.5.2.2 UePower

class UePower

UePower commands group definition. 11 total commands, 4 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.tpc.carrier.uePower.clone()
```

Subgroups

7.5.2.2.1 Maximum

SCPI Commands

```

FETCh:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:UEPower:MAXimum
READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:UEPower:MAXimum
CALCulate:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:UEPower: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'
- Ue_Power: float: float UE power Range: -100 dBm to 55 dBm, Unit: dBm
- Max_Output_Power: float: float Maximum output power Range: -100 dBm to 55 dBm, Unit: dBm
- Min_Outpu_Power: float: float Minimum output power Range: -100 dBm to 55 dBm, Unit: dBm

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Ue_Power: float: float UE power Range: -100 dBm to 55 dBm, Unit: dBm
- Max_Output_Power: float: float Maximum output power Range: -100 dBm to 55 dBm, Unit: dBm
- Min_Outpu_Power: float: float Minimum output power Range: -100 dBm to 55 dBm, Unit: dBm

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```

# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:UEPower:MAXimum
value: CalculateStruct = driver.tpc.carrier.uePower.maximum.calculate(carrier =
↳repcap.Carrier.Default)

```

Return the UE power and minimum/maximum output power single value results per carrier. The minimum, maximum and average values of these results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```

# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:MAXimum
value: ResultData = driver.tpc.carrier.uePower.maximum.fetch(carrier = repcap.
↳Carrier.Default)

```

(continues on next page)

(continued from previous page)

Return the UE power and minimum/maximum output power single value results per carrier. The minimum, maximum and average values of these results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return structure: for return value, see the help for ResultData structure arguments.

read(*carrier=<Carrier.Default: -1>*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:MAXimum
value: ResultData = driver.tpc.carrier.uePower.maximum.read(carrier = repcap.
↳Carrier.Default)
```

Return the UE power and minimum/maximum output power single value results per carrier. The minimum, maximum and average values of these results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return structure: for return value, see the help for ResultData structure arguments.

7.5.2.2.2 Minimum

SCPI Commands

```
FETCh:WCDMA:MEASurement<Instance>:TPC:CARRIER<Carrier>:UEPower:MINimum
READ:WCDMA:MEASurement<Instance>:TPC:CARRIER<Carrier>:UEPower:MINimum
CALCulate:WCDMA:MEASurement<Instance>:TPC:CARRIER<Carrier>:UEPower: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'
- Ue_Power: float: float UE power Range: -100 dBm to 55 dBm, Unit: dBm
- Max_Output_Power: float: float Maximum output power Range: -100 dBm to 55 dBm, Unit: dBm
- Min_Outpu_Power: float: float Minimum output power Range: -100 dBm to 55 dBm, Unit: dBm

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Ue_Power: float: float UE power Range: -100 dBm to 55 dBm, Unit: dBm
- Max_Output_Power: float: float Maximum output power Range: -100 dBm to 55 dBm, Unit: dBm
- Min_Outpu_Power: float: float Minimum output power Range: -100 dBm to 55 dBm, Unit: dBm

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:UEPower:MINimum
value: CalculateStruct = driver.tpc.carrier.uePower.minimum.calculate(carrier =
↳repcap.Carrier.Default)
```

Return the UE power and minimum/maximum output power single value results per carrier. The minimum, maximum and average values of these results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:MINimum
value: ResultData = driver.tpc.carrier.uePower.minimum.fetch(carrier = repcap.
↳Carrier.Default)
```

Return the UE power and minimum/maximum output power single value results per carrier. The minimum, maximum and average values of these results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

read(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:MINimum
value: ResultData = driver.tpc.carrier.uePower.minimum.read(carrier = repcap.
↳Carrier.Default)
```

Return the UE power and minimum/maximum output power single value results per carrier. The minimum, maximum and average values of these results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

7.5.2.2.3 Average

SCPI Commands

```

FETCH:WCDMA:MEASUREMENT<Instance>:TPC:CARRIER<Carrier>:UEPower:AVERAGE
READ:WCDMA:MEASUREMENT<Instance>:TPC:CARRIER<Carrier>:UEPower:AVERAGE
CALCULATE:WCDMA:MEASUREMENT<Instance>:TPC:CARRIER<Carrier>:UEPower: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’
- Ue_Power: float: float UE power Range: -100 dBm to 55 dBm, Unit: dBm
- Max_Output_Power: float: float Maximum output power Range: -100 dBm to 55 dBm, Unit: dBm
- Min_Outpu_Power: float: float Minimum output power Range: -100 dBm to 55 dBm, Unit: dBm

class ResultData

Response structure. Fields:

- Reliability: int: decimal ‘Reliability Indicator’
- Ue_Power: float: float UE power Range: -100 dBm to 55 dBm, Unit: dBm
- Max_Output_Power: float: float Maximum output power Range: -100 dBm to 55 dBm, Unit: dBm
- Min_Outpu_Power: float: float Minimum output power Range: -100 dBm to 55 dBm, Unit: dBm

calculate(*carrier*=<Carrier.Default: -1>) → CalculateStruct

```

# SCPI: CALCULATE:WCDMA:MEASUREMENT<instance>:TPC:CARRIER<carrier>
↳:UEPower:AVERAGE
value: CalculateStruct = driver.tpc.carrier.uePower.average.calculate(carrier =
↳repcap.Carrier.Default)

```

Return the UE power and minimum/maximum output power single value results per carrier. The minimum, maximum and average values of these results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(*carrier*=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:AVERage
value: ResultData = driver.tpc.carrier.uePower.average.fetch(carrier = repcap.
↳Carrier.Default)
```

Return the UE power and minimum/maximum output power single value results per carrier. The minimum, maximum and average values of these results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return structure: for return value, see the help for ResultData structure arguments.

read(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:AVERage
value: ResultData = driver.tpc.carrier.uePower.average.read(carrier = repcap.
↳Carrier.Default)
```

Return the UE power and minimum/maximum output power single value results per carrier. The minimum, maximum and average values of these results can be retrieved. The command returns all parameters listed below, independent of the selected TPC setup. However, only for some of the parameters measured values are available. For the other parameters, only an indicator is returned (e.g. NAV) . 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.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return structure: for return value, see the help for ResultData structure arguments.

7.5.2.2.4 Statistics

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:UEPower:STATistics
READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:UEPower:STATistics
```

class Statistics

Statistics commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Max_Output_Power: int: decimal Number of trace values for maximum output power Range: 0 to 341
- Min_Outpu_Power: int: decimal Number of trace values for minimum output power Range: 0 to 341

fetch(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:UEPower:STATistics
value: ResultData = driver.tpc.carrier.uePower.statistics.fetch(carrier =
↳repcap.Carrier.Default)
```

Return the ‘Statistics’ values, indicating how many trace values have been considered to derive the results. The results are the maximum, minimum and average values of the maximum output power and the minimum output power per carrier. The command returns all parameters listed below, independent of the selected TPC setup. Depending on the TPC setup, either a result value or an indicator is returned (e.g. NAV) .

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

read(carrier=<Carrier.Default: -1>) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:STATistics
value: ResultData = driver.tpc.carrier.uePower.statistics.read(carrier = repcap.
↳Carrier.Default)
```

Return the ‘Statistics’ values, indicating how many trace values have been considered to derive the results. The results are the maximum, minimum and average values of the maximum output power and the minimum output power per carrier. The command returns all parameters listed below, independent of the selected TPC setup. Depending on the TPC setup, either a result value or an indicator is returned (e.g. NAV) .

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Carrier’)

return structure: for return value, see the help for ResultData structure arguments.

7.5.2.3 Trace

class Trace

Trace 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.tpc.carrier.trace.clone()
```

Subgroups

7.5.2.3.1 UePower

class UePower

UePower 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.tpc.carrier.trace.uePower.clone()
```

Subgroups

7.5.2.3.1.1 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:TRACe:UEPower:CURRent
READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:TRACe:UEPower:CURRent
CALCulate:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:TRACe:UEPower:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:TRACe:UEPower:CURRent
value: List[float] = driver.tpc.carrier.trace.uePower.current.calculate(carrier,
↳repcap.Carrier.Default)
```

Return the values of the UE power vs slot trace per carrier. You can query the number of measured slots using the CONFigure:WCDMa:MEAS:TPC:...:MLENth? command of the used measurement mode. 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 RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return ue_power: float N power results, one per measured slot Range: -100 dBm to 55 dBm, Unit: dBm

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:TRACe:UEPower:CURRent
value: List[float] = driver.tpc.carrier.trace.uePower.current.fetch(carrier =
↳repcap.Carrier.Default)
```

Return the values of the UE power vs slot trace per carrier. You can query the number of measured slots using the CONFigure:WCDMa:MEAS:TPC:...:MLENth? command of the used measurement mode. 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 RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return ue_power: float N power results, one per measured slot Range: -100 dBm to 55 dBm, Unit: dBm

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:TRACe:UEPower:CURRent
value: List[float] = driver.tpc.carrier.trace.uePower.current.read(carrier =
↳repcap.Carrier.Default)
```

Return the values of the UE power vs slot trace per carrier. You can query the number of measured slots using the `CONFigure:WCDMa:MEAS:TPC:...:MLENth?` command of the used measurement mode. 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 `RsCmwWcdmaMeas.reliability.last_value` to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return ue_power: float N power results, one per measured slot Range: -100 dBm to 55 dBm, Unit: dBm

7.5.2.3.2 Psteps

class Psteps

Psteps 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.tpc.carrier.trace.psteps.clone()
```

Subgroups

7.5.2.3.2.1 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:TRACe:PSTeps:CURRent
READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:TRACe:PSTeps:CURRent
CALCulate:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:TRACe:PSTeps:CURRent
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:TRACe:PSTeps:CURRent
value: List[float] = driver.tpc.carrier.trace.psteps.current.calculate(carrier =
↳repcap.Carrier.Default)
```

Return the values of the power steps trace per carrier. Each power step is calculated as the difference between the UE power of a slot and the UE power of the preceding slot. For the first measured slot, a 0 is returned. You can query the number of measured slots using the CONFIGure:WCDMa:MEAS:TPC:...:MLENgt? command of the used measurement mode. 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 RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return power_steps: float N power step results, one per measured slot Power step result number m indicates the difference between the UE power results number m and number m-1. The first power step result equals NCAP. Range: -50 dB to 50 dB, Unit: dB

fetch(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:TRACe:PSTeps:CURRent
value: List[float] = driver.tpc.carrier.trace.psteps.current.fetch(carrier =
↳repcap.Carrier.Default)
```

Return the values of the power steps trace per carrier. Each power step is calculated as the difference between the UE power of a slot and the UE power of the preceding slot. For the first measured slot, a 0 is returned. You can query the number of measured slots using the CONFIGure:WCDMa:MEAS:TPC:...:MLENgt? command of the used measurement mode. 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 RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return power_steps: float N power step results, one per measured slot Power step result number m indicates the difference between the UE power results number m and number m-1. The first power step result equals NCAP. Range: -50 dB to 50 dB, Unit: dB

read(carrier=<Carrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:TRACe:PSTeps:CURRent
value: List[float] = driver.tpc.carrier.trace.psteps.current.read(carrier =
↳repcap.Carrier.Default)
```

Return the values of the power steps trace per carrier. Each power step is calculated as the difference between the UE power of a slot and the UE power of the preceding slot. For the first measured slot, a 0 is returned. You can query the number of measured slots using the CONFIGure:WCDMa:MEAS:TPC:...:MLENgt? command of the used measurement mode. 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 RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param carrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

return power_steps: float N power step results, one per measured slot Power step result number m indicates the difference between the UE power results number m and number m-1. The first power step result equals NCAP. Range: -50 dB to 50 dB, Unit: dB

7.5.3 Total

class Total

Total 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.tpc.total.clone()
```

Subgroups

7.5.3.1 UePower

class UePower

UePower 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.tpc.total.uePower.clone()
```

Subgroups

7.5.3.1.1 Maximum

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:TPC:TOTal:UEPower:MAXimum
READ:WCDMa:MEASurement<Instance>:TPC:TOTal:UEPower:MAXimum
```

class Maximum

Maximum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Ue_Power: float: float UE power Range: -100 dBm to 55 dBm, Unit: dBm
- Max_Output_Power: float: float Maximum output power Range: -100 dBm to 55 dBm, Unit: dBm

fetch() → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MAXimum
value: ResultData = driver.tpc.total.uePower.maximum.fetch()
```

Return the UE power and maximum output power single value results over all carriers. The minimum, maximum and average values of these results can be retrieved.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MAXimum
value: ResultData = driver.tpc.total.uePower.maximum.read()
```

Return the UE power and maximum output power single value results over all carriers. The minimum, maximum and average values of these results can be retrieved.

return structure: for return value, see the help for ResultData structure arguments.

7.5.3.1.2 Minimum

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:TPC:TOTal:UEPower:MINimum
READ:WCDMA:MEASurement<Instance>:TPC:TOTal:UEPower:MINimum
```

class Minimum

Minimum commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Ue_Power: float: float UE power Range: -100 dBm to 55 dBm, Unit: dBm
- Max_Output_Power: float: float Maximum output power Range: -100 dBm to 55 dBm, Unit: dBm

fetch() → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MINimum
value: ResultData = driver.tpc.total.uePower.minimum.fetch()
```

Return the UE power and maximum output power single value results over all carriers. The minimum, maximum and average values of these results can be retrieved.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MINimum
value: ResultData = driver.tpc.total.uePower.minimum.read()
```


Return the UE power and maximum output power single value results over all carriers. The minimum, maximum and average values of these results can be retrieved.

return structure: for return value, see the help for ResultData structure arguments.

7.5.3.1.3 Average

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:TPC:TOTal:UEPower:AVERage
READ:WCDMA:MEASurement<Instance>:TPC:TOTal:UEPower:AVERage
```

class Average

Average commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Ue_Power: float: float UE power Range: -100 dBm to 55 dBm, Unit: dBm
- Max_Output_Power: float: float Maximum output power Range: -100 dBm to 55 dBm, Unit: dBm

fetch() → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:AVERage
value: ResultData = driver.tpc.total.uePower.average.fetch()
```

Return the UE power and maximum output power single value results over all carriers. The minimum, maximum and average values of these results can be retrieved.

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:AVERage
value: ResultData = driver.tpc.total.uePower.average.read()
```

Return the UE power and maximum output power single value results over all carriers. The minimum, maximum and average values of these results can be retrieved.

return structure: for return value, see the help for ResultData structure arguments.

7.5.3.1.4 Statistics

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:TPC:TOTal:UEPower:STATistics
READ:WCDMA:MEASurement<Instance>:TPC:TOTal:UEPower:STATistics
```

class Statistics

Statistics commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → int

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:TOTal:UEPower:STATistics  
value: int = driver.tpc.total.uePower.statistics.fetch()
```

Return the ‘Statistics’ values, indicating how many trace values have been considered to derive the maximum, minimum and average values of the maximum output power over all carriers.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return max_output_power: decimal Number of trace values for maximum output power over all carriers Range: 0 to 341

read() → int

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:TOTal:UEPower:STATistics  
value: int = driver.tpc.total.uePower.statistics.read()
```

Return the ‘Statistics’ values, indicating how many trace values have been considered to derive the maximum, minimum and average values of the maximum output power over all carriers.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return max_output_power: decimal Number of trace values for maximum output power over all carriers Range: 0 to 341

7.5.3.2 Trace

class Trace

Trace 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.tpc.total.trace.clone()
```

Subgroups

7.5.3.2.1 UePower

class UePower

UePower 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.tpc.total.trace.uePower.clone()
```

Subgroups

7.5.3.2.1.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:TPC:TOTal:TRACe:UEPower:CURRent
FETCh:WCDMa:MEASurement<Instance>:TPC:TOTal:TRACe:UEPower:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:TOTal:TRACe:UEPower:CURRent
value: List[float] = driver.tpc.total.trace.uePower.current.fetch()
```

Return the values of the UE power vs slot trace over all carriers. You can query the number of measured slots using the CONFigure:WCDMa:MEAS:TPC:...:MLENgtH? command of the used measurement mode.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float N power results, one per measured slot Range: -100 dBm to 55 dBm, Unit: dBm

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:TOTal:TRACe:UEPower:CURRent
value: List[float] = driver.tpc.total.trace.uePower.current.read()
```

Return the values of the UE power vs slot trace over all carriers. You can query the number of measured slots using the CONFigure:WCDMa:MEAS:TPC:...:MLENgtH? command of the used measurement mode.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float N power results, one per measured slot Range: -100 dBm to 55 dBm, Unit: dBm

7.5.4 State

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:TPC:STATe
```

class State

State commands group definition. 2 total commands, 1 Sub-groups, 1 group commands

fetch() → RsCmwWcdmaMeas.enums.ResourceState

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:TPC:STATe
value: enums.ResourceState = driver.tpc.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 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.tpc.state.clone()
```

Subgroups

7.5.4.1 All

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:TPC: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:WCDMa:MEASurement<instance>:TPC:STAtE:ALL
value: FetchStruct = driver.tpc.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.

7.6 Prach

SCPI Commands

```
STOP:WCDMa:MEASurement<Instance>:PRACH
ABORT:WCDMa:MEASurement<Instance>:PRACH
INITiate:WCDMa:MEASurement<Instance>:PRACH
```

class Prach

Prach commands group definition. 39 total commands, 4 Sub-groups, 3 group commands

abort() → None

```
# SCPI: ABORT:WCDMa:MEASurement<instance>:PRACH
driver.prach.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:WCDMa:MEASurement<instance>:PRACH
driver.prach.abort_with_opc()

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.

Same as abort, but waits for the operation to complete before continuing further. Use the RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

initiate() → None

```

# SCPI: INITiate:WCDMa:MEASurement<instance>:PRACH
driver.prach.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:WCDMa:MEASurement<instance>:PRACH
driver.prach.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 RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

stop() → None

```
# SCPI: STOP:WCDma:MEASurement<instance>:PRCh
driver.prach.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.
- 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:WCDma:MEASurement<instance>:PRCh
driver.prach.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 RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.prach.clone()
```

Subgroups

7.6.1 State

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:PRACH:STATe
```

class State

State commands group definition. 2 total commands, 1 Sub-groups, 1 group commands

fetch() → RsCmwWcdmaMeas.enums.ResourceState

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:PRACH:STATe
value: enums.ResourceState = driver.prach.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 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.prach.state.clone()
```

Subgroups

7.6.1.1 All

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:PRACH: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:WCDMa:MEASurement<instance>:PRACH:STATe:ALL
value: FetchStruct = driver.prach.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.

7.6.2 Trace

class Trace

Trace commands group definition. 28 total commands, 7 Sub-groups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.prach.trace.clone()
```

Subgroups

7.6.2.1 UePower

class UePower

UePower 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.prach.trace.uePower.clone()
```

Subgroups

7.6.2.1.1 Current

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:PRACH:TRACe:UEPower:CURRent
FETCh:WCDMA:MEASurement<Instance>:PRACH:TRACe:UEPower:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:PRACH:TRACe:UEPower:CURRent
value: List[float] = driver.prach.trace.uePower.current.fetch()
```

Return the values of the UE power bar graph. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: -100 dBm to 55 dBm, Unit: dBm

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:PRACH:TRACe:UEPower:CURRent
value: List[float] = driver.prach.trace.uePower.current.read()
```

Return the values of the UE power bar graph. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: -100 dBm to 55 dBm, Unit: dBm

7.6.2.1.2 Chip

class Chip

Chip 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.prach.trace.uePower.chip.clone()
```

Subgroups

7.6.2.1.2.1 Current

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:PRACH:TRACe:UEPower:CHIP:CURRent
FETCh:WCDMA:MEASurement<Instance>:PRACH:TRACe:UEPower:CHIP:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:PRACH:TRACe:UEPower:CHIP:CURRent
value: List[float] = driver.prach.trace.uePower.chip.current.fetch()
```

Return the values of the UE power vs. chip diagram. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power_chip: float Comma-separated list of 9216 values, one per chip: 2560 values before last preamble, 4096 values for preselected preamble, 2560 values after last preamble Range: -100 dBm to 55 dBm, Unit: dBm

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:PRACH:TRACe:UEPower:CHIP:CURRent
value: List[float] = driver.prach.trace.uePower.chip.current.read()
```

Return the values of the UE power vs. chip diagram. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return ue_power_chip: float Comma-separated list of 9216 values, one per chip: 2560 values before last preamble, 4096 values for preselected preamble, 2560 values after last preamble Range: -100 dBm to 55 dBm, Unit: dBm

7.6.2.2 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.prach.trace.evMagnitude.clone()
```

Subgroups

7.6.2.2.1 Rms

class Rms

Rms 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.prach.trace.evMagnitude.rms.clone()
```

Subgroups

7.6.2.2.1.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:EVMagnitude:RMS:CURRENT
FETCh:WCDMa:MEASurement<Instance>:PRACH:TRACe:EVMagnitude:RMS:CURRENT
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude[:RMS]:CURRENT
value: List[float] = driver.prach.trace.evMagnitude.rms.current.fetch()
```

Return the EVM RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm: float Comma-separated list of values, one result per measured preamble
(see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: 0 % to 100
%, Unit: %

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude[:RMS]:CURRENT
value: List[float] = driver.prach.trace.evMagnitude.rms.current.read()
```

Return the EVM RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm: float Comma-separated list of values, one result per measured preamble
(see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: 0 % to 100
%, Unit: %

7.6.2.2.2 Peak

class Peak

Peak 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.prach.trace.evMagnitude.peak.clone()
```

Subgroups

7.6.2.2.2.1 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:PRACH:TRACe:EVMagnitude:PEAK:CURRent
READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:EVMagnitude:PEAK:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude:PEAK:CURRent
value: List[float] = driver.prach.trace.evMagnitude.peak.current.fetch()
```

Return the EVM RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm: float Comma-separated list of values, one result per measured preamble
(see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: 0 % to 100
%, Unit: %

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude:PEAK:CURRent
value: List[float] = driver.prach.trace.evMagnitude.peak.current.read()
```

Return the EVM RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm: float Comma-separated list of values, one result per measured preamble
(see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: 0 % to 100
%, Unit: %

7.6.2.2.3 Chip

class Chip

Chip 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.prach.trace.evMagnitude.chip.clone()
```

Subgroups

7.6.2.2.3.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:EVMagnitude:CHIP:CURRent
FETCh:WCDMa:MEASurement<Instance>:PRACH:TRACe:EVMagnitude:CHIP:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude:CHIP:CURRent
value: List[float] = driver.prach.trace.evMagnitude.chip.current.fetch()
```

Return the values of the error vector magnitude vs. chip diagram. See also ‘Detailed Views: Modulation’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_chip: float Comma-separated list of 4096 values, one per chip of the pre-selected preamble Range: 0 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude:CHIP:CURRent
value: List[float] = driver.prach.trace.evMagnitude.chip.current.read()
```

Return the values of the error vector magnitude vs. chip diagram. See also ‘Detailed Views: Modulation’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return evm_chip: float Comma-separated list of 4096 values, one per chip of the pre-selected preamble Range: 0 % to 100 %, Unit: %

7.6.2.3 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.prach.trace.merror.clone()
```

Subgroups

7.6.2.3.1 Rms

class Rms

Rms 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.prach.trace.merror.rms.clone()
```

Subgroups

7.6.2.3.1.1 Current

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:PRACH:TRACe:MERRor:RMS:CURRent
READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:MERRor:RMS:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:PRACH:TRACe:MERRor[:RMS]:CURRent
value: List[float] = driver.prach.trace.merror.rms.current.fetch()
```

Return the magnitude error RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return magnitude_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: PEAK: -100 % to 100 %, RMS: 0 % to 100 % , Unit: %

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:PRCh:TRAcE:MERRor[:RMS]:CURRent
value: List[float] = driver.prach.trace.merror.rms.current.read()
```

Return the magnitude error RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return magnitude_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: PEAK: -100 % to 100 %, RMS: 0 % to 100 % , Unit: %

7.6.2.3.2 Peak

class Peak

Peak 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.prach.trace.merror.peak.clone()
```

Subgroups

7.6.2.3.2.1 Current

SCPI Commands

```
FEtCh:WCDma:MEASurement<Instance>:PRCh:TRAcE:MERRor:PEAK:CURRent
READ:WCDma:MEASurement<Instance>:PRCh:TRAcE:MERRor:PEAK:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FEtCh:WCDma:MEASurement<instance>:PRCh:TRAcE:MERRor:PEAK:CURRent
value: List[float] = driver.prach.trace.merror.peak.current.fetch()
```

Return the magnitude error RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return magnitude_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: PEAK: -100 % to 100 %, RMS: 0 % to 100 % , Unit: %

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:PRCh:TRAcE:MERRor:PEAK:CURRent
value: List[float] = driver.prach.trace.merror.peak.current.read()
```


Return the magnitude error RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return magnitude_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: PEAK: -100 % to 100 %, RMS: 0 % to 100 % , Unit: %

7.6.2.3.3 Chip

class Chip

Chip 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.prach.trace.merror.chip.clone()
```

Subgroups

7.6.2.3.3.1 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:PRACH:TRACe:MERRor:CHIP:CURRent
READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:MERRor:CHIP:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:MERRor:CHIP:CURRent
value: List[float] = driver.prach.trace.merror.chip.current.fetch()
```

Return the values of the magnitude error vs. chip diagram. See also ‘Detailed Views: Modulation’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_chip: float Comma-separated list of 4096 values, one per chip of the preselected preamble Range: -100 % to 100 %, Unit: %

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:TRACe:MERRor:CHIP:CURRent
value: List[float] = driver.prach.trace.merror.chip.current.read()
```

Return the values of the magnitude error vs. chip diagram. See also ‘Detailed Views: Modulation’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return mag_error_chip: float Comma-separated list of 4096 values, one per chip of the preselected preamble Range: -100 % to 100 %, Unit: %

7.6.2.4 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.prach.trace.perror.clone()
```

Subgroups

7.6.2.4.1 Rms

class Rms

Rms 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.prach.trace.perror.rms.clone()
```

Subgroups

7.6.2.4.1.1 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:PRACH:TRACe:PERRor:RMS:CURRent
READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:PERRor:RMS:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:PERRor[:RMS]:CURRent
value: List[float] = driver.prach.trace.perror.rms.current.fetch()
```

Return the phase error RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: PEAK: -180 deg to 180 deg, RMS: 0 deg to 180 deg , Unit: deg

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:PRACH:TRACe:PERRor[:RMS]:CURRent
value: List[float] = driver.prach.trace.perror.rms.current.read()
```

Return the phase error RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: PEAK: -180 deg to 180 deg, RMS: 0 deg to 180 deg , Unit: deg

7.6.2.4.2 Peak

class Peak

Peak 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.prach.trace.perror.peak.clone()
```

Subgroups

7.6.2.4.2.1 Current

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:PRACH:TRACe:PERRor:PEAK:CURRent
READ:WCDma:MEASurement<Instance>:PRACH:TRACe:PERRor:PEAK:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:PRACH:TRACe:PERRor:PEAK:CURRent
value: List[float] = driver.prach.trace.perror.peak.current.fetch()
```

Return the phase error RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: PEAK: -180 deg to 180 deg, RMS: 0 deg to 180 deg , Unit: deg

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:PRACH:TRACe:PERRor:PEAK:CURRent
value: List[float] = driver.prach.trace.perror.peak.current.read()
```

Return the phase error RMS and peak values for each measured preamble.

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: PEAK: -180 deg to 180 deg, RMS: 0 deg to 180 deg , Unit: deg

7.6.2.4.3 Chip

class Chip

Chip 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.prach.trace.perror.chip.clone()
```

Subgroups

7.6.2.4.3.1 Current

SCPI Commands

```
FETCH:WCDMa:MEASurement<Instance>:PRACH:TRACe:PERRor:CHIP:CURRent
READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:PERRor:CHIP:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:PERRor:CHIP:CURRent
value: List[float] = driver.prach.trace.perror.chip.current.fetch()
```

Return the values of the phase error vs. chip diagram. See also ‘Detailed Views: Modulation’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_chip: float Comma-separated list of 4096 values, one per chip of the preselected preamble Range: -180 deg to 180 deg, Unit: deg

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:TRACe:PERRor:CHIP:CURRent
value: List[float] = driver.prach.trace.perror.chip.current.read()
```

Return the values of the phase error vs. chip diagram. See also ‘Detailed Views: Modulation’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return phase_error_chip: float Comma-separated list of 4096 values, one per chip of the preselected preamble Range: -180 deg to 180 deg, Unit: deg

7.6.2.5 FreqError

class FreqError

FreqError 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.prach.trace.freqError.clone()
```

Subgroups

7.6.2.5.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:FERRor:CURRent
FETCh:WCDMa:MEASurement<Instance>:PRACH:TRACe:FERRor:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:PRACH:TRACe:FERRor:CURRent
value: List[float] = driver.prach.trace.freqError.current.fetch()
```

Return the values of the frequency error bar graph. See also ‘Detailed Views: Modulation’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return frequency_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: -60000 Hz to 60000 Hz, Unit: Hz

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:TRACe:FERRor:CURRent
value: List[float] = driver.prach.trace.freqError.current.read()
```

Return the values of the frequency error bar graph. See also ‘Detailed Views: Modulation’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return frequency_error: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) Range: -60000 Hz to 60000 Hz, Unit: Hz

7.6.2.6 Psteps

class Psteps

Psteps 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.prach.trace.psteps.clone()
```

Subgroups

7.6.2.6.1 Current

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:PRACH:TRACe:PSTeps:CURRent
FETCh:WCDMA:MEASurement<Instance>:PRACH:TRACe:PSTeps:CURRent
```

class Current

Current commands group definition. 2 total commands, 0 Sub-groups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:PRACH:TRACe:PSTeps:CURRent
value: List[float] = driver.prach.trace.psteps.current.fetch()
```

Return the values of the power steps bar graph. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return power_steps: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) For the first preamble NCAP is returned. Range: -10 dB to 50 dB, Unit: dB

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:PRACH:TRACe:PSTeps:CURRent
value: List[float] = driver.prach.trace.psteps.current.read()
```

Return the values of the power steps bar graph. See also ‘Detailed Views: UE Power and Power Steps’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return power_steps: float Comma-separated list of values, one result per measured preamble (see method RsCmwWcdmaMeas.Configure.Prach.mpreamble) For the first preamble NCAP is returned. Range: -10 dB to 50 dB, Unit: dB

7.6.2.7 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.prach.trace.iq.clone()
```

Subgroups

7.6.2.7.1 Current

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:IQ:CURRent
FETCh:WCDMa:MEASurement<Instance>:PRACH: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]: float I amplitude of a constellation point Range: -5 to 5
- Qphase: List[float]: float Q amplitude of a constellation point Range: -5 to 5

fetch() → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:PRACH:TRACe:IQ:CURRent
value: ResultData = driver.prach.trace.iq.current.fetch()
```

Returns the results in the I/Q constellation diagram, see also 'Detailed Views: I/Q Constellation Diagram'. The constellation points are returned as pairs of I and Q values: <Reliability>, <Iphase>1, <Qphase>1, ..., <Iphase>3904, <Qphase>3904

return structure: for return value, see the help for ResultData structure arguments.

read() → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:TRACe:IQ:CURRent
value: ResultData = driver.prach.trace.iq.current.read()
```

Returns the results in the I/Q constellation diagram, see also 'Detailed Views: I/Q Constellation Diagram'. The constellation points are returned as pairs of I and Q values: <Reliability>, <Iphase>1, <Qphase>1, ..., <Iphase>3904, <Qphase>3904

return structure: for return value, see the help for ResultData structure arguments.

7.6.3 OffPower

SCPI Commands

```

FETCh:WCDMa:MEASurement<Instance>:PRACH:OFFPower
READ:WCDMa:MEASurement<Instance>:PRACH:OFFPower
CALCulate:WCDMa:MEASurement<Instance>:PRACH:OFFPower

```

class OffPower

OffPower commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

calculate() → List[float]

```

# SCPI: CALCulate:WCDMa:MEASurement<instance>:PRACH:OFFPower
value: List[float] = driver.prach.offPower.calculate()

```

Return the OFF power results. See also ‘Detailed Views: TX Measurement’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return off_power: float OFF power before preamble, OFF power after preamble Range:
-100 dBm to -24 dBm, Unit: dBm

fetch() → List[float]

```

# SCPI: FETCh:WCDMa:MEASurement<instance>:PRACH:OFFPower
value: List[float] = driver.prach.offPower.fetch()

```

Return the OFF power results. See also ‘Detailed Views: TX Measurement’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return off_power: float OFF power before preamble, OFF power after preamble Range:
-100 dBm to -24 dBm, Unit: dBm

read() → List[float]

```

# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:OFFPower
value: List[float] = driver.prach.offPower.read()

```

Return the OFF power results. See also ‘Detailed Views: TX Measurement’

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

return off_power: float OFF power before preamble, OFF power after preamble Range:
-100 dBm to -24 dBm, Unit: dBm

7.6.4 Preamble<Preamble>

RepCap Settings

```
# Range: Nr1 .. Nr5
rc = driver.prach.preamble.repcap_preamble_get()
driver.prach.preamble.repcap_preamble_set(repcap.Preamble.Nr1)
```

class Preamble

Preamble commands group definition. 3 total commands, 1 Sub-groups, 0 group commands Repeated Capability: Preamble, default value after init: Preamble.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.prach.preamble.clone()
```

Subgroups

7.6.4.1 Current

SCPI Commands

```
READ:WCDMA:MEASurement<Instance>:PRACH:PREamble<Preamble>:CURRENT
FETCh:WCDMA:MEASurement<Instance>:PRACH:PREamble<Preamble>:CURRENT
CALCulate:WCDMA:MEASurement<Instance>:PRACH:PREamble<Preamble>:CURRENT
```

class Current

Current commands group definition. 3 total commands, 0 Sub-groups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Ue_Power: float: float Mean preamble power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float Mean preamble power minus mean power of previous preamble For first preamble NCAP is returned. Range: -10 dB to 50 dB, Unit: dB
- Carrier_Freq_Err: float: float Carrier frequency error Range: -60000 Hz to 60000 Hz, Unit: Hz
- 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: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 %, Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Ue_Power: float: float Mean preamble power Range: -100 dBm to 55 dBm, Unit: dBm
- Power_Steps: float: float Mean preamble power minus mean power of previous preamble For first preamble NCAP is returned. Range: -10 dB to 50 dB, Unit: dB
- Carrier_Freq_Err: float: float Carrier frequency error Range: -60000 Hz to 60000 Hz, Unit: Hz
- 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: %
- Mag_Error_Rms: float: float Magnitude error RMS value Range: 0 % to 100 %, Unit: %
- Mag_Error_Peak: float: float Magnitude error peak value Range: -100 % to 100 %, Unit: %
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- 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
- Signature: int: decimal Detected preamble signature Range: 0 to 15

calculate(preamble=<Preamble.Default: -1>) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:PRCh:PREamble<nr>:CURRent
value: CalculateStruct = driver.prach.preamble.current.calculate(preamble = repcap.Preamble.Default)
```

Return the single value results for a selected preamble. See also 'Detailed Views: TX Measurement'

Use RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param preamble optional repeated capability selector. Default value: Nr1 (settable in the interface 'Preamble')

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch(preamble=<Preamble.Default: -1>) → ResultData

```
# SCPI: FETCh:WCDma:MEASurement<instance>:PRCh:PREamble<nr>:CURRent
value: ResultData = driver.prach.preamble.current.fetch(preamble = repcap.Preamble.Default)
```

Return the single value results for a selected preamble. See also 'Detailed Views: TX Measurement'

param preamble optional repeated capability selector. Default value: Nr1 (settable in the interface 'Preamble')

return structure: for return value, see the help for ResultData structure arguments.

read(preamble=<Preamble.Default: -1>) → ResultData

```
# SCPI: READ:WCDma:MEASurement<instance>:PRACH:PREamble<nr>:CURRENT
value: ResultData = driver.prach.preamble.current.read(preamble = repcap.
↳Preamble.Default)
```

Return the single value results for a selected preamble. See also ‘Detailed Views: TX Measurement’

param preamble optional repeated capability selector. Default value: Nr1 (settable in the interface ‘Preamble’)

return structure: for return value, see the help for ResultData structure arguments.

7.7 OoSync

SCPI Commands

```
CALCulate:WCDma:MEASurement<Instance>:OOSync
READ:WCDma:MEASurement<Instance>:OOSync
FETCh:WCDma:MEASurement<Instance>:OOSync
STOP:WCDma:MEASurement<Instance>:OOSync
ABORt:WCDma:MEASurement<Instance>:OOSync
INITiate:WCDma:MEASurement<Instance>:OOSync
```

class OoSync

OoSync commands group definition. 8 total commands, 1 Sub-groups, 6 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal See ‘Reliability Indicator’
- Out_Pow_Ab_Max: enums.ResultStatus2: float Maximal output power measured in interval A-B Unit: dBm
- Out_Pow_Ab_Min: enums.ResultStatus2: float Minimal output power measured in interval A-B Unit: dBm
- Out_Pow_Ccurrent: enums.ResultStatus2: float Output power measured for point C Unit: dBm
- Out_Pow_Cd_Max: enums.ResultStatus2: float Maximal output power measured in interval C-D Unit: dBm
- Out_Pow_Cd_Min: enums.ResultStatus2: float Minimal output power measured in interval C-D Unit: dBm
- Out_Pow_De_Max: enums.ResultStatus2: float Maximal output power measured in interval D-E Unit: dBm
- Out_Pow_De_Min: enums.ResultStatus2: float Minimal output power measured in interval D-E Unit: dBm
- Out_Pow_Fcurrent: enums.ResultStatus2: float Output power measured for point F Unit: dBm

class ResultData

Response structure. Fields:

- Reliability: int: decimal See ‘Reliability Indicator’
- Out_Pow_Ab_Max: float: float Maximal output power measured in interval A-B Unit: dBm

- Out_Pow_Ab_Min: float: float Minimal output power measured in interval A-B Unit: dBm
- Out_Pow_Ccurrent: float: float Output power measured for point C Unit: dBm
- Out_Powc_State: enums.OutPowFstate: OFF | NOFF State of output power for point C OFF: UE transmitter off NOFF: UE transmitter not off
- Out_Pow_Cd_Max: float: float Maximal output power measured in interval C-D Unit: dBm
- Out_Pow_Cd_Min: float: float Minimal output power measured in interval C-D Unit: dBm
- Out_Pow_De_Max: float: float Maximal output power measured in interval D-E Unit: dBm
- Out_Pow_De_Min: float: float Minimal output power measured in interval D-E Unit: dBm
- Out_Pow_Fcurrent: float: float Output power measured for point F Unit: dBm
- Out_Pow_Fstate: enums.OutPowFstate: ON | NON State of output power for point F ON: UE transmitter on NON: UE transmitter not on

abort() → None

```
# SCPI: ABORT:WCDMA:MEASurement<instance>:OOSync
driver.ooSync.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:WCDMA:MEASurement<instance>:OOSync
driver.ooSync.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 RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:OOSync
value: CalculateStruct = driver.ooSync.calculate()
```

Return the results of out-of-synchronization handling measurement.

return structure: for return value, see the help for CalculateStruct structure arguments.

fetch() → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:OOSync
value: ResultData = driver.ooSync.fetch()
```

Return the results of out-of-synchronization handling measurement.

return structure: for return value, see the help for ResultData structure arguments.

initiate() → None

```
# SCPI: INITiate:WCDMa:MEASurement<instance>:OOSync
driver.ooSync.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:WCDMa:MEASurement<instance>:OOSync
driver.ooSync.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.

(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.

Same as initiate, but waits for the operation to complete before continuing further. Use the RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

read() → ResultData

```

# SCPI: READ:WCDMa:MEASurement<instance>:OOSync
value: ResultData = driver.ooSync.read()

```

Return the results of out-of-synchronization handling measurement.

return structure: for return value, see the help for ResultData structure arguments.

stop() → None

```

# SCPI: STOP:WCDMa:MEASurement<instance>:OOSync
driver.ooSync.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.
- 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:WCDMa:MEASurement<instance>:OOSync
driver.ooSync.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.

```

(continues on next page)

(continued from previous page)

Use `FETCh...STATe?` to query the current measurement state.

Same as `stop`, but waits for the operation to complete before continuing further. Use the `RsCmwWcdmaMeas.utilities.opc_timeout_set()` to set the timeout value.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.ooSync.clone()
```

Subgroups

7.7.1 State

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:OOSync:STATe
```

class State

State commands group definition. 2 total commands, 1 Sub-groups, 1 group commands

fetch() → `RsCmwWcdmaMeas.enums.ResourceState`

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:OOSync:STATe
value: enums.ResourceState = driver.ooSync.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 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.ooSync.state.clone()
```

Subgroups

7.7.1.1 All

SCPI Commands

```
FETCH:WCDma:MEASurement<Instance>:OOSync: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:WCDma:MEASurement<instance>:OOSync:STATe:ALL
value: FetchStruct = driver.ooSync.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.

7.8 OlpControl

SCPI Commands

```
STOP:WCDma:MEASurement<Instance>:OLPControl
ABORt:WCDma:MEASurement<Instance>:OLPControl
INITiate:WCDma:MEASurement<Instance>:OLPControl
READ:WCDma:MEASurement<Instance>:OLPControl
FETCh:WCDma:MEASurement<Instance>:OLPControl
CALCulate:WCDma:MEASurement<Instance>:OLPControl
```

class OlpControl

OlpControl commands group definition. 10 total commands, 2 Sub-groups, 6 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Olpc_1: float: float UE power in DPCCH power control preamble of carrier one during measurement of the ramp up of carrier one Range: -100 dBm to 100 dBm
- Olpc_2: float: float UE power in DPCCH power control preamble of carrier two during measurement of the ramp up of carrier two Range: -100 dBm to 100 dBm

class ResultData

Response structure. Fields:

- Reliability: int: decimal 'Reliability Indicator'
- Ue_Pwr_C_1: float: float UE power of carrier one during measurement of the ramp up of carrier two Range: -100 dBm to 100 dBm
- Olpc_1: float: float UE power in DPCCH power control preamble of carrier one during measurement of the ramp up of carrier one Range: -100 dBm to 100 dBm
- Slot_No_C_1: int: decimal Slot where the power ramp up of carrier one has been detected Range: 0 slots to 14 slots
- Olpc_2: float: float UE power in DPCCH power control preamble of carrier two during measurement of the ramp up of carrier two Range: -100 dBm to 100 dBm
- Slot_No_C_2: int: decimal Slot where the power ramp up of carrier two has been detected Range: 0 slots to 14 slots

abort() → None

```
# SCPI: ABORT:WCDMA:MEASurement<instance>:OLPControl
driver.olpControl.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:WCDMA:MEASurement<instance>:OLPControl
driver.olpControl.abort_with_opc()

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.

Same as abort, but waits for the operation to complete before continuing further. Use the RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

calculate() → CalculateStruct

```

# SCPI: CALCulate:WCDMa:MEASurement<instance>:OLPControl
value: CalculateStruct = driver.olpControl.calculate()

```

Return the single value results for open loop power control measurements. 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:WCDMa:MEASurement<instance>:OLPControl
value: ResultData = driver.olpControl.fetch()

```

Return the single value results for open loop power control measurements. 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.

initiate() → None

```

# SCPI: INITiate:WCDMa:MEASurement<instance>:OLPControl
driver.olpControl.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:WCDMa:MEASurement<instance>:OLPControl
driver.olpControl.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 RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

read() → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:OLPControl
value: ResultData = driver.olpControl.read()
```

Return the single value results for open loop power control measurements. 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.

stop() → None

```
# SCPI: STOP:WCDMa:MEASurement<instance>:OLPControl
driver.olpControl.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.
- 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:WCDMA:MEASurement<instance>:OLPControl
driver.olpControl.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 RsCmwWcdmaMeas.utilities.opc_timeout_set() to set the timeout value.

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.olpControl.clone()
```

Subgroups

7.8.1 State

SCPI Commands

```
FETCH:WCDMA:MEASurement<Instance>:OLPControl:STATE
```

class State

State commands group definition. 2 total commands, 1 Sub-groups, 1 group commands

fetch() → RsCmwWcdmaMeas.enums.ResourceState

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:OLPControl:STATE
value: enums.ResourceState = driver.olpControl.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 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.olpControl.state.clone()
```

Subgroups

7.8.1.1 All

SCPI Commands

```
FETCh:WCDMa:MEASurement<Instance>:OLPControl: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:WCDMa:MEASurement<instance>:OLPControl:STATE:ALL
value: FetchStruct = driver.olpControl.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.

7.8.2 Carrier<CARRierExt>

RepCap Settings

```
# Range: Nr1 .. Nr32
rc = driver.olpControl.carrier.repcap_cARRierExt_get()
driver.olpControl.carrier.repcap_cARRierExt_set(repcap.CARRierExt.Nr1)
```

class Carrier

Carrier commands group definition. 2 total commands, 1 Sub-groups, 0 group commands Repeated Capability: CARRierExt, default value after init: CARRierExt.Nr1

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.olpControl.carrier.clone()
```

Subgroups**7.8.2.1 UepPower****class UepPower**

UepPower 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.olpControl.carrier.uepPower.clone()
```

Subgroups**7.8.2.1.1 Rup<RampUpCarrier>****RepCap Settings**

```
# Range: Nr1 .. Nr32
rc = driver.olpControl.carrier.uepPower.rup.repcap_rampUpCarrier_get()
driver.olpControl.carrier.uepPower.rup.repcap_rampUpCarrier_set(repcap.RampUpCarrier.Nr1)
```

SCPI Commands

```
READ:WCDMa:MEASurement<Instance>:OLPControl:CARRier<CARRierExt>:UEPPower:RUP
↳<RampUpCarrier>
FETCh:WCDMa:MEASurement<Instance>:OLPControl:CARRier<CARRierExt>:UEPPower:RUP
↳<RampUpCarrier>
```

class Rup

Rup commands group definition. 2 total commands, 0 Sub-groups, 2 group commands Repeated Capability: RampUpCarrier, default value after init: RampUpCarrier.Nr1

fetch(cARRierExt=<CARRierExt.Default: -1>, rampUpCarrier=<RampUpCarrier.Default: -1>) →
List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:OLPControl:CARRier<carrier>
↳:UEPPower:RUP<rupcarrier>
value: List[float] = driver.olpControl.carrier.uepPower.rup.fetch(cARRierExt =
↳repcap.CARRierExt.Default, rampUpCarrier = repcap.RampUpCarrier.Default)
```

Return the traces of the UE power vs slot during the ramp up of selected carrier per uplink carrier measured in slots -15 to 45. 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 RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param cARRierExt optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param rampUpCarrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Rup')

return ue_power: float 60 UE power results, one per measured slot Range: -100 dBm to 100 dBm , Unit: dBm

read(cARRierExt=<CARRierExt.Default: -1>, rampUpCarrier=<RampUpCarrier.Default: -1>) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:OLPControl:CARRier<carrier>
↳:UEPPower:RUP<rupcarrier>
value: List[float] = driver.olpControl.carrier.uepPower.rup.read(cARRierExt =
↳repcap.CARRierExt.Default, rampUpCarrier = repcap.RampUpCarrier.Default)
```

Return the traces of the UE power vs slot during the ramp up of selected carrier per uplink carrier measured in slots -15 to 45. 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 RsCmwWcdmaMeas.reliability.last_value to read the updated reliability indicator.

param cARRierExt optional repeated capability selector. Default value: Nr1 (settable in the interface 'Carrier')

param rampUpCarrier optional repeated capability selector. Default value: Nr1 (settable in the interface 'Rup')

return ue_power: float 60 UE power results, one per measured slot Range: -100 dBm to 100 dBm , Unit: dBm

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.olpControl.carrier.uepPower.rup.clone()
```


INDEX

A

ABORT:WCDma:MEASurement<Instance>:MEValuation,
168
ABORT:WCDma:MEASurement<Instance>:OLPControl,
552
ABORT:WCDma:MEASurement<Instance>:OOSync, 547
ABORT:WCDma:MEASurement<Instance>:PRACH, 525
ABORT:WCDma:MEASurement<Instance>:TPC, 492

C

CALCulate:WCDma:MEASurement<Instance>:MEValuation: 332
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 330
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 328
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 325
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 336
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 334
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 338
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 340
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 368
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 366
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 357
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 360
CALCulate:WCDma:MEASurement<Instance>:MEValuation: 363
CALCulate:WCDma:MEASurement<Instance>:OLPControl, 552
CALCulate:WCDma:MEASurement<Instance>:OOSync, 547
CALCulate:WCDma:MEASurement<Instance>:PRACH:OFFPower, 544
CALCulate:WCDma:MEASurement<Instance>:PRACH:PREamble<Pream 545
CALCulate:WCDma:MEASurement<Instance>:TPC:CARRIER<Carrier> 505
CALCulate:WCDma:MEASurement<Instance>:TPC:CARRIER<Carrier> 500
CALCulate:WCDma:MEASurement<Instance>:TPC:CARRIER<Carrier> 503
CALCulate:WCDma:MEASurement<Instance>:TPC:CARRIER<Carrier> 517
CALCulate:WCDma:MEASurement<Instance>:TPC:CARRIER<Carrier> 516
CALCulate:WCDma:MEASurement<Instance>:TPC:CARRIER<Carrier> 513
CALCulate:WCDma:MEASurement<Instance>:TPC:CARRIER<Carrier> 510
CALCulate:WCDma:MEASurement<Instance>:TPC:CARRIER<Carrier> 511
CALCulate:WCDma:MEASurement<Instance>:TPC:DHIB:AVERage, 497
CALCulate:WCDma:MEASurement<Instance>:TPC:DHIB:MAXimum, 495
CALCulate:WCDma:MEASurement<Instance>:TPC:DHIB:MINimumc, 499
CONFIGure:WCDma:MEASurement<Instance>:CARRIER<Carrier>:BAN 44
CONFIGure:WCDma:MEASurement<Instance>:CELL:CARRIER<Carrier>:MODulation:PHDhsdpccch, 45
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:AMODE:MO 92
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:CDTHresh 91
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:DMODE:MO 92
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:DSFactor 90
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:LIMit:AC 88
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:LIMit:AC 88
CONFIGure:WCDma:MEASurement<Instance>:MEValuation:LIMit:CF 68

CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:LIST:SEQ
 85 106
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurementABSDistance<Instance>:MEValuation:LIST:SEQ
 87 110
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:LIST:SEQ
 85 105
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:LIST:SEQ
 68 107
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:LIST:SEQ
 68 109
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:MOEXcept
 68 63
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:MPERiod:
 68 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:MSCount,
 84 63
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:PSLot,
 84 63
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:REPetiti
 68 63
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:A
 68 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:A
 68 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:B
 72 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:C
 74 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:C
 83 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:C
 82 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:C
 79 101
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:C
 77 101
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:C
 80 101
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:C
 81 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:E
 103 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:E
 112 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:I
 103 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:M
 103 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:F
 103 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:F
 108 93
 CONFIGure:WCDma:MEASurement<Instance>:MEValuatCONFigme::WCDma:MEASurement<Instance>:MEValuation:RESult:F
 111 93

CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:LIMit:PControl
 93 138
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:LIMit:PControl
 93 136
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:LIMit:PControl
 93 136
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:LIMit:PControl
 89 136
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:LIMit:PERror,
 63 133
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:MOEXception,
 66 131
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:MPReamble,
 66 131
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:OFFPower,
 66 131
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:PPReamble,
 90 131
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:ALL,
 63 141
 CONFIGure:WCDma:MEASurement<Instance>:MEValueatCONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:CHIP:EV
 63 145
 CONFIGure:WCDma:MEASurement<Instance>:OLPControl:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:CHIP:ME
 149 145
 CONFIGure:WCDma:MEASurement<Instance>:OLPControl:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:CHIP:PE
 149 145
 CONFIGure:WCDma:MEASurement<Instance>:OLPControl:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:CHIP:UE
 149 145
 CONFIGure:WCDma:MEASurement<Instance>:OOSync:ADDPCh:level:WCDma:MEASurement<Instance>:PRACH:RESult:EVMagni
 147 141
 CONFIGure:WCDma:MEASurement<Instance>:OOSync:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:FERRor,
 148 141
 CONFIGure:WCDma:MEASurement<Instance>:OOSync:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:IQ,
 148 141
 CONFIGure:WCDma:MEASurement<Instance>:OOSync:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:MERRor,
 148 141
 CONFIGure:WCDma:MEASurement<Instance>:OOSync:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:PERror,
 147 141
 CONFIGure:WCDma:MEASurement<Instance>:PRACH:LIMit:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:PSTeps,
 133 141
 CONFIGure:WCDma:MEASurement<Instance>:PRACH:LIMit:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:RESult:UEPower
 133 141
 CONFIGure:WCDma:MEASurement<Instance>:PRACH:LIMit:CONF:RESult:WCDma:MEASurement<Instance>:PRACH:TOUT,
 133 131
 CONFIGure:WCDma:MEASurement<Instance>:PRACH:LIMit:CONF:RESult:WCDma:MEASurement<Instance>:RFSettings:CARRier<C
 133 62
 CONFIGure:WCDma:MEASurement<Instance>:PRACH:LIMit:CONF:RESult:WCDma:MEASurement<Instance>:RFSettings:DCARRier:
 133 61
 CONFIGure:WCDma:MEASurement<Instance>:PRACH:LIMit:CONF:RESult:WCDma:MEASurement<Instance>:RFSettings:EATTenuat
 138 59
 CONFIGure:WCDma:MEASurement<Instance>:PRACH:LIMit:CONF:RESult:WCDma:MEASurement<Instance>:RFSettings:ENPower,
 138 59
 CONFIGure:WCDma:MEASurement<Instance>:PRACH:LIMit:CONF:RESult:WCDma:MEASurement<Instance>:RFSettings:UMARgin,
 138 59

```

CONFIGure:WCDma:MEASurement<Instance>:TPC:CSELEnabled:WCDma:MEASurement<Instance>:TPC:MPEDch:MLENgtH,
113 118
CONFIGure:WCDma:MEASurement<Instance>:TPC:CTFCOFF:WCDma:MEASurement<Instance>:TPC:SETup,
119 113
CONFIGure:WCDma:MEASurement<Instance>:TPC:DHIBON:WCDma:MEASurement<Instance>:TPC:TOUT,
121 113
CONFIGure:WCDma:MEASurement<Instance>:TPC:DHIBON:WCDma:MEASurement<Instance>:TPC:ULCM:AEExecution,
121 120
CONFIGure:WCDma:MEASurement<Instance>:TPC:DHIBON:WCDma:MEASurement<Instance>:TPC:ULCM:MLENgtH,
121 120
CONFIGure:WCDma:MEASurement<Instance>:TPC:ILPControl:WCDma:MEASurement<Instance>:UEChannels:BSFSelect
115 50
CONFIGure:WCDma:MEASurement<Instance>:TPC:ILPControl:WCDma:MEASurement<Instance>:UEChannels:CARRier<
115 57
CONFIGure:WCDma:MEASurement<Instance>:TPC:ILPControl:WCDma:MEASurement<Instance>:UEChannels:CARRier<
115 56
CONFIGure:WCDma:MEASurement<Instance>:TPC:ILPControl:WCDma:MEASurement<Instance>:UEChannels:CARRier<
115 53
CONFIGure:WCDma:MEASurement<Instance>:TPC:ILPControl:WCDma:MEASurement<Instance>:UEChannels:CARRier<
115 51
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:WCDma:MEASurement<Instance>:UEChannels:CARRier<
122 54
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:WCDma:MEASurement<Instance>:UEChannels:CARRier<
122 55
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:WCDma:MEASurement<Instance>:UESignal:CARRier<Car
124 49
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:WCDma:MEASurement<Instance>:UESignal:CMPattern,
127 46
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:WCDma:MEASurement<Instance>:UESignal:DPDCh,
127 46
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:WCDma:MEASurement<Instance>:UESignal:SFormat,
127 46
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:WCDma:MEASurement<Instance>:UESignal:ULConfig,
127 46
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:ILPControl:MINPower,
124 F
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:TPControl:MEASGroup:MEValuation:BER,
124 369
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:TPControl:MEASGroup:MEValuation:CARRier<Car
124 353
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:TPControl:MEASGroup:MEValuation:CARRier<Car
122 352
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:TPControl:MEASGroup:MEValuation:CARRier<Car
129 354
CONFIGure:WCDma:MEASurement<Instance>:TPC:LIMit:TPControl:MEASGroup:MEValuation:CARRier<Car
129 355
CONFIGure:WCDma:MEASurement<Instance>:TPC:MODEFETCh:WCDma:MEASurement<Instance>:MEValuation:CARRier<Car
113 346
CONFIGure:WCDma:MEASurement<Instance>:TPC:MOEXception:WCDma:MEASurement<Instance>:MEValuation:CARRier<Car
113 345
CONFIGure:WCDma:MEASurement<Instance>:TPC:MONitor:WCDma:MEASurement<Instance>:MEValuation:CARRier<Car
118 349
CONFIGure:WCDma:MEASurement<Instance>:TPC:MPEDch:WCDma:MEASurement<Instance>:MEValuation:CARRier<Car
118 348

```


FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:CDPower:ISDPeak:Minimum:CARRIER<Carrier>
 251 233
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:CDPower:ISDPeak:SDENat:CARRIER<Carrier>
 253 231
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:EMStat:code:MPEAK:AVERAGE:CARRIER<Carrier>
 219 230
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:EMStat:code:MPEAK:CURRENT:CARRIER<Carrier>
 220 295
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:EMStat:code:MPEAK:MAXIMUM:CARRIER<Carrier>
 217 296
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:EMStat:code:MPEAK:SDENat:CARRIER<Carrier>
 218 294
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:EMStat:code:MPS:AVERAGE:CARRIER<Carrier>
 215 293
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:EMStat:code:MPS:CURRENT:CARRIER<Carrier>
 216 292
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:EMStat:code:MPS:MAXIMUM:CARRIER<Carrier>
 214 304
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:EMStat:code:MPS:SDENat:CARRIER<Carrier>
 213 303
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:FPS:code:AVERAGE:evaluation:CARRIER<Carrier>
 290 305
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:FPS:code:CURRENT:evaluation:CARRIER<Carrier>
 291 306
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:FPS:code:MAXIMUM:evaluation:CARRIER<Carrier>
 289 309
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:FPS:code:SDENat:evaluation:CARRIER<Carrier>
 288 308
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:MPS:code:PEAK:AVERAGE:region:CARRIER<Carrier>
 228 310
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:MPS:code:PEAK:CURRENT:region:CARRIER<Carrier>
 229 311
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:MPS:code:PEAK:MAXIMUM:region:CARRIER<Carrier>
 227 317
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:MPS:code:PEAK:SDENat:region:CARRIER<Carrier>
 226 316
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:MPS:code:MPS:AVERAGE:region:CARRIER<Carrier>
 224 318
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:MPS:code:MPS:CURRENT:region:CARRIER<Carrier>
 225 319
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:MPS:code:MPS:MAXIMUM:region:CARRIER<Carrier>
 223 322
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:MPS:code:MPS:SDENat:region:CARRIER<Carrier>
 222 321
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:PPS:code:PEAK:AVERAGE:region:CARRIER<Carrier>
 236 323
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:PPS:code:PEAK:CURRENT:region:CARRIER<Carrier>
 237 324
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:PPS:code:PEAK:MAXIMUM:region:CARRIER<Carrier>
 235 313
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:PPS:code:PEAK:SDENat:region:CARRIER<Carrier>
 234 312
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER:WCDMA:MEASurement:PACKET:PPS:code:MPS:AVERAGE:region:CARRIER<Carrier>
 232 314

FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGMENT
 472 376
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGMENT
 473 377
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGMENT
 472 375
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGMENT
 473 384
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:MEASurement<Instance>:MEvaluation:LIST:SEGMENT
 474 383
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:CODEASUREMENT<Instance>:MEvaluation:LIST:SEGMENT
 397 386
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:CODEASUREMENT<Instance>:MEvaluation:LIST:SEGMENT
 396 374
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:CURREASUREMENT<Instance>:MEvaluation:LIST:SPECTRUM
 400 441
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:CURREASUREMENT<Instance>:MEvaluation:LIST:SPECTRUM
 399 440
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:CURREASUREMENT<Instance>:MEvaluation:LIST:SPECTRUM
 399 441
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:MAXASUREMENT<Instance>:MEvaluation:LIST:SPECTRUM
 400 443
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:PHMEASUREMENT<Instance>:MEvaluation:LIST:SPECTRUM
 398 442
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:PHMEASUREMENT<Instance>:MEvaluation:LIST:SPECTRUM
 397 444
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH WCDMA:PHMEASUREMENT<Instance>:MEvaluation:LIST:SPECTRUM
 395 447
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:CDPower:CDPowerAVERAGEvaluation:LIST:SPECTRUM
 392 438
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:CDPower:CDPowerCURRENTvaluation:LIST:SPECTRUM
 391 438
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:CDPower:CDPowerMAXIMUMvaluation:LIST:SPECTRUM
 393 439
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:CDPower:CDPowerSDEvaluation:LIST:SPECTRUM
 394 446
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:CDPower:CDPowerAVERAGEvaluation:LIST:SPECTRUM
 379 425
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:CDPower:CDPowerCURRENTvaluation:LIST:SPECTRUM
 378 424
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:CDPower:CDPowerMAXIMUMvaluation:LIST:SPECTRUM
 381 426
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:CDPower:CDPowerMINIMUMvaluation:LIST:SPECTRUM
 380 437
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:CDPower:CDPowerSDEvaluation:LIST:SPECTRUM
 382 436
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:MODulation:AVERagevaluation:LIST:SPECTRUM
 390 437
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:MODulation:CURRENTvaluation:LIST:SPECTRUM
 389 427
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:MODulation:MAXIMUMvaluation:LIST:SPECTRUM
 388 426
 FETCH:WCDMA:MEASurement<Instance>:MEvaluation:FFSCH SEGMENT:MEASUREMENT:MODulation:SDEvaluation:LIST:SPECTRUM
 387 427

FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:LIST:SPECTrum:AVErAge
 435 420
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:LIST:SRELiab
 435 372
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:LIST:UEPower
 436 373
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:MODulation:P
 428 368
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:MODulation:U
 428 366
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:PCDE:CURRENT
 429 370
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:PCDE:MAXimum
 433 371
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:SPECTrum:AVErAge
 433 357
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:SPECTrum:CURR
 434 360
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:SPECTrum:MAX
 430 363
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:STATE,
 430 171
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:STATE:ALL,
 431 172
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:CDEMOR
 432 176
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:CDEMOR
 431 175
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:CDPMOR
 432 178
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:CDPMOR
 423 177
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 423 188
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 424 189
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 422 190
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 421 191
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 422 192
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 449 193
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 445 195
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 445 194
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 446 196
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 420 181
 FETCH:WCDMA:MEASurement<Instance>:MEValuation:FHSC:SPECtrum:MEASurement<Instance>:MEValuation:TRACE:EMASK:
 419 182

Index	571
--------------	------------

524 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 FETCH:WCDMA:MEASurement<Instance>:TPC:STATE:ALL, 336
 524 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 FETCH:WCDMA:MEASurement<Instance>:TPC:TOTAL:TRACE:UEPower:CURRENT, 334
 523 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 FETCH:WCDMA:MEASurement<Instance>:TPC:TOTAL:UEPower:AVERAGE, 335
 521 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 FETCH:WCDMA:MEASurement<Instance>:TPC:TOTAL:UEPower:MAXIMUM, 336
 519 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 FETCH:WCDMA:MEASurement<Instance>:TPC:TOTAL:UEPower:MINIMUM, 337
 520 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 FETCH:WCDMA:MEASurement<Instance>:TPC:TOTAL:UEPower:STATISTICS, 338
 521 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 I 267
 INITiate:WCDMA:MEASurement<Instance>:MEvaluation, 266
 168 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 INITiate:WCDMA:MEASurement<Instance>:OLPControl, 268
 552 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 INITiate:WCDMA:MEASurement<Instance>:OOSync, 269
 547 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 INITiate:WCDMA:MEASurement<Instance>:PRACH, 271
 525 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>
 INITiate:WCDMA:MEASurement<Instance>:TPC, 492 270
 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier> 272
 READ:WCDMA:MEASurement<Instance>:MEvaluation:BER, 273
 369 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:AVERAGE, 280
 353 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:CURRENT, 279
 352 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:MAXIMUM, 281
 354 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:MINIMUM, 282
 355 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:SDDeviation, 284
 346 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:AVERAGE, 283
 345 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:CURRENT, 285
 349 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:MAXIMUM, 287
 348 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:MINIMUM, 275
 350 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:CDPower:SDDeviation, 274
 332 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:MODulation:AVERAGE, 276
 330 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:MODulation:CURRENT, 277
 328 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:MODulation:MAXIMUM, 277
 325 READ:WCDMA:MEASurement<Instance>:MEvaluation:CARRIER<Carrier>:MODulation:SDDeviation, 239

READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:MEvaluation:Carrier<Carrier>
 293 210
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:SEMEvaluation:Carrier<Carrier>
 292 211
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:DPCKVal:Average;Carrier<Carrier>
 304 209
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:DPCKVal:Current;Carrier<Carrier>
 303 208
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:DPCKVal:Maximum;Carrier<Carrier>
 305 207
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:DPCKVal:SEDeviation;Modulation:PH
 306 368
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:DPCKVal:Average;Modulation:UE
 309 366
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:DPCKVal:Current;PCDE:CURRENT;
 308 370
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:DPCKVal:Maximum;PCDE:MAXIMUM;
 310 371
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:DPCKVal:SEDeviation;Spectrum:AVE
 311 357
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:EDPCKVal:Average;Spectrum:CURR
 317 360
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:EDPCKVal:Current;Spectrum:MAXI
 316 363
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:EDPCKVal:Maximum;TRACE:Carrier
 318 180
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:EDPCKVal:SEDeviation;TRACE:CDEMONI
 319 176
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:EDPCKVal:FastFlop;TRACE:OVERRANG
 322 175
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:EDPCKVal:FastFlop;TRACE:CDPRERMI
 321 178
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:EDPCKVal:FastFlop;TRACE:CDPRMOMI
 323 177
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:EDPCKVal:FastFlop;TRACE:SEMSWIA
 324 188
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:HSDPVal:Average;TRACE:EMASK:F
 313 189
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:HSDPVal:Current;TRACE:EMASK:F
 312 190
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:HSDPVal:Maximum;TRACE:EMASK:F
 314 191
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:HSDPVal:SEDeviation;TRACE:EMASK:F
 315 192
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:SFMEPCh:TRACE:EMASK:F
 298 193
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:SFMEPCh:TRACE:EMASK:F
 299 195
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:SFMEPCh:TRACE:EMASK:F
 301 194
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:SFMEPCh:FastFlop;TRACE:EMASK:F
 302 196
 READ:WCDMa:MEASurement<Instance>:MEvaluation:Carrier<Carrier>:PSTStep:SFMEPCh:TRACE:EMASK:F
 300 181

READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EMASk:MFLeft:CURRent,
 182 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:MERRor:RMS:CU
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EMASk:MFLeft:MAXimum,
 183 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:PERRor:CHIP:Q
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EMASk:MFRight:AVERAge,
 184 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:PERRor:PEAK:Q
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EMASk:MFRight:CURRent,
 186 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:PERRor:RMS:CU
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EMASk:MFRight:MAXimum,
 187 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:PSTeps:CURRer
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EVMAgnitude:CHIP:AVERAge,
 198 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:UEPower:CHIP:Q
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EVMAgnitude:CHIP:CURRent,
 199 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:UEPower:CURRer
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:EVMAgnitude:CHIP:MAXimum,
 197 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTe
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:IQ5CURRent,
 205 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTe
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:MERRor:CHIP:AVERAge,
 201 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTe
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:MERRor:CHIP:CURRent,
 202 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:PSTe
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:MERRor:CHIP:MAXimum,
 200 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:TRAC
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:PERRor:CHIP:AVERAge,
 204 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:TRAC
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:PERRor:CHIP:CURRent,
 204 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:UEPo
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:PERRor:CHIP:MAXimum,
 203 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:UEPo
 READ:WCDMa:MEASurement<Instance>:MEvaluation:TRACe:PHD5CURRent,
 173 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:UEPo
 READ:WCDMa:MEASurement<Instance>:OLPControl, 511
 552 READ:WCDMa:MEASurement<Instance>:TPC:CARRier<Carrier>:UEPo
 READ:WCDMa:MEASurement<Instance>:OLPControl:CARRier<CarrierExt>:UEPPower:RUP<RampUpCarrier>,
 558 READ:WCDMa:MEASurement<Instance>:TPC:DHIB:AVERAge,
 READ:WCDMa:MEASurement<Instance>:OOSync, 547 497
 READ:WCDMa:MEASurement<Instance>:PRACH:OFFPower, 495
 544 READ:WCDMa:MEASurement<Instance>:TPC:DHIB:MAXimum,
 READ:WCDMa:MEASurement<Instance>:PRACH:PREamble:PRACH:MEASurement<Instance>:TPC:DHIB:MINimum,
 545 496
 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:EVMAgnitude:CHIP:CURRent, 498
 534 READ:WCDMa:MEASurement<Instance>:TPC:DHIB:STATistics,
 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:EVMAgnitude:PEAK:CURRent, 523
 533 READ:WCDMa:MEASurement<Instance>:TPC:TOTal:TRACe:UEPower:Q
 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:EVMAgnitude:RMS:CURRent, 521
 532 READ:WCDMa:MEASurement<Instance>:TPC:TOTal:UEPower:AVERAge
 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:FERor:WCDMA:MEASurement<Instance>:TPC:TOTal:UEPower:MAXimum
 541 519
 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:IQCDPRM, 520
 543 READ:WCDMa:MEASurement<Instance>:TPC:TOTal:UEPower:MINimum
 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:MERRor:WCDMA:MEASurement<Instance>:TPC:TOTal:UEPower:STATist
 537 521
 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:MERRor:PEAK:CURRent, 41
 READ:WCDMa:MEASurement<Instance>:PRACH:TRACe:MERRor:PEAK:CURRent, 41

ROUTe:WCDMa:MEASurement<Instance>:SCENario, 41 TRIGger:WCDMa:MEASurement<Instance>:OOSync:MGAP, 162
 ROUTe:WCDMa:MEASurement<Instance>:SCENario:CSPT, 41 TRIGger:WCDMa:MEASurement<Instance>:OOSync:SLOPe, 162
 ROUTe:WCDMa:MEASurement<Instance>:SCENario:MAP, 43 TRIGger:WCDMa:MEASurement<Instance>:OOSync:SOURce, 162
 ROUTe:WCDMa:MEASurement<Instance>:SCENario:SALT, 41 TRIGger:WCDMa:MEASurement<Instance>:OOSync:THReshold, 162
 S TRIGger:WCDMa:MEASurement<Instance>:OOSync:TOUT, 162
 STOP:WCDMa:MEASurement<Instance>:MEValuation, 168 TRIGger:WCDMa:MEASurement<Instance>:PRACH:CATalog:SOURce, 161
 STOP:WCDMa:MEASurement<Instance>:OLPControl, 552 TRIGger:WCDMa:MEASurement<Instance>:PRACH:DElay, 159
 STOP:WCDMa:MEASurement<Instance>:OOSync, 547 TRIGger:WCDMa:MEASurement<Instance>:PRACH:MGAP, 159
 STOP:WCDMa:MEASurement<Instance>:PRACH, 525 TRIGger:WCDMa:MEASurement<Instance>:PRACH:SLOPe, 159
 STOP:WCDMa:MEASurement<Instance>:TPC, 492 TRIGger:WCDMa:MEASurement<Instance>:PRACH:SOURce, 159
 T TRIGger:WCDMa:MEASurement<Instance>:MEValuation:CATalog:SOURce, 159
 TRIGger:WCDMa:MEASurement<Instance>:MEValuation:DElay, 151 TRIGger:WCDMa:MEASurement<Instance>:PRACH:THReshold, 159
 TRIGger:WCDMa:MEASurement<Instance>:MEValuation:LIST:MODE, 154 TRIGger:WCDMa:MEASurement<Instance>:TPC:CATalog:SOURce, 159
 TRIGger:WCDMa:MEASurement<Instance>:MEValuation:MGAP, 151 TRIGger:WCDMa:MEASurement<Instance>:TPC:DElay, 155
 TRIGger:WCDMa:MEASurement<Instance>:MEValuation:SLOPe, 151 TRIGger:WCDMa:MEASurement<Instance>:TPC:MGAP, 155
 TRIGger:WCDMa:MEASurement<Instance>:MEValuation:SOURce, 151 TRIGger:WCDMa:MEASurement<Instance>:TPC:SLOPe, 155
 TRIGger:WCDMa:MEASurement<Instance>:MEValuation:THReshold, 151 TRIGger:WCDMa:MEASurement<Instance>:TPC:SOURce, 155
 TRIGger:WCDMa:MEASurement<Instance>:MEValuation:TOUT, 151 TRIGger:WCDMa:MEASurement<Instance>:TPC:THReshold, 155
 TRIGger:WCDMa:MEASurement<Instance>:OLPControl:CATalog:SOURce, 168 TRIGger:WCDMa:MEASurement<Instance>:TPC:TOUT, 155
 TRIGger:WCDMa:MEASurement<Instance>:OLPControl:DElay, 165
 TRIGger:WCDMa:MEASurement<Instance>:OLPControl:MGAP, 165
 TRIGger:WCDMa:MEASurement<Instance>:OLPControl:SLOPe, 165
 TRIGger:WCDMa:MEASurement<Instance>:OLPControl:SOURce, 165
 TRIGger:WCDMa:MEASurement<Instance>:OLPControl:THReshold, 165
 TRIGger:WCDMa:MEASurement<Instance>:OLPControl:TOUT, 165
 TRIGger:WCDMa:MEASurement<Instance>:OOSync:CATalog:SOURce, 165
 TRIGger:WCDMa:MEASurement<Instance>:OOSync:DElay, 162