
RsCMPX_WcdmaMeas

Release 5.0.40.3

Rohde & Schwarz

Apr 19, 2024

CONTENTS:

1	Revision History	3
1.1	RsCMPX_WcdmaMeas	3
1.1.1	Version history	3
2	Getting Started	5
2.1	Introduction	5
2.2	Installation	7
2.3	Finding Available Instruments	8
2.4	Initiating Instrument Session	9
2.5	Plain SCPI Communication	12
2.6	Error Checking	14
2.7	Exception Handling	14
2.8	Transferring Files	16
2.9	Writing Binary Data	16
2.10	Transferring Big Data with Progress	17
2.11	Multithreading	18
2.12	Logging	21
3	Enums	25
3.1	AcIrMode	25
3.2	ActiveLimit	25
3.3	AnalysisMode	25
3.4	AutoManualMode	25
3.5	Band	26
3.6	Carrier	26
3.7	CmwsConnector	26
3.8	DetectionMode	27
3.9	LimitHmode	27
3.10	MeasMode	27
3.11	MeasPeriod	27
3.12	Mode	27
3.13	Modulation	28
3.14	OutPowFstate	28
3.15	ParameterSetMode	28
3.16	PatternType	28
3.17	PcdErrorPhase	28
3.18	Repeat	29
3.19	ResourceState	29
3.20	ResultStatus2	29
3.21	Retrigger	29

3.22	RxConnector	29
3.23	RxConverter	30
3.24	SetType	30
3.25	SignalSlope	31
3.26	SlotNumber	31
3.27	SpreadingFactorA	31
3.28	SpreadingFactorB	31
3.29	State	31
3.30	StopCondition	32
3.31	TargetMainState	32
3.32	TargetSyncState	32
3.33	TestCase	32
3.34	TestScenarioB	32
3.35	Type	33
3.36	UIConfiguration	33
4	RepCaps	35
4.1	Instance (Global)	35
4.2	Carrier	35
4.3	CARRierExt	35
4.4	EdpdChannel	36
4.5	Minus	36
4.6	Plus	36
4.7	Preamble	36
4.8	RampUpCarrier	36
4.9	Segment	37
5	Examples	39
6	RsCMPX_WcdmaMeas API Structure	41
6.1	Configure	44
6.1.1	WcdmaMeas	44
6.1.1.1	Carrier<Carrier>	44
6.1.1.1.1	Band	45
6.1.1.2	Cell	46
6.1.1.2.1	Carrier<Carrier>	46
6.1.1.2.1.1	Scode	46
6.1.1.3	MultiEval	47
6.1.1.3.1	Amode	50
6.1.1.3.2	CdThreshold	51
6.1.1.3.3	Dmode	52
6.1.1.3.4	DsFactor	52
6.1.1.3.5	Limit	53
6.1.1.3.5.1	Aclr	54
6.1.1.3.5.2	Relative	55
6.1.1.3.5.3	Emask	56
6.1.1.3.5.4	Absolute	57
6.1.1.3.5.5	Dcarrier	58
6.1.1.3.5.6	Absolute	58
6.1.1.3.5.7	EvMagnitude	59
6.1.1.3.5.8	Merror	60
6.1.1.3.5.9	Pcontrol	61
6.1.1.3.5.10	EpStep	61
6.1.1.3.5.11	Hsdpcch	62

6.1.1.3.5.12	Perror	63
6.1.1.3.5.13	Phd	64
6.1.1.3.5.14	PhsDpcch	65
6.1.1.3.5.15	RcdError	66
6.1.1.3.5.16	EecdP	67
6.1.1.3.5.17	Carrier<Carrier>	67
6.1.1.3.5.18	Dpcch	71
6.1.1.3.5.19	Dpdch	72
6.1.1.3.5.20	Edpcch	74
6.1.1.3.5.21	Edpdch<EdpdChannel>	75
6.1.1.3.5.22	Hsdpcch	76
6.1.1.3.5.23	Config	78
6.1.1.3.6	ListPy	79
6.1.1.3.6.1	Segment<Segment>	81
6.1.1.3.6.2	CdPower	82
6.1.1.3.6.3	Cidx	83
6.1.1.3.6.4	Modulation	84
6.1.1.3.6.5	Phd	85
6.1.1.3.6.6	Setup	86
6.1.1.3.6.7	SingleCmw	88
6.1.1.3.6.8	Connector	88
6.1.1.3.6.9	Spectrum	89
6.1.1.3.6.10	UePower	90
6.1.1.3.6.11	SingleCmw	91
6.1.1.3.7	Mperiod	91
6.1.1.3.8	Result	92
6.1.1.3.8.1	Chip	102
6.1.1.3.9	Rotation	104
6.1.1.3.10	Scount	104
6.1.1.3.11	Sscalar	106
6.1.1.4	OlpControl	106
6.1.1.5	OoSync	108
6.1.1.5.1	Limit	109
6.1.1.6	Prach	110
6.1.1.6.1	Limit	112
6.1.1.6.1.1	EvMagnitude	114
6.1.1.6.1.2	Merror	115
6.1.1.6.1.3	Pcontrol	115
6.1.1.6.1.4	MaxPower	116
6.1.1.6.1.5	UserDefined	118
6.1.1.6.1.6	OlPower	119
6.1.1.6.1.7	Pstep	120
6.1.1.6.1.8	Perror	120
6.1.1.6.2	Result	121
6.1.1.6.2.1	Chip	126
6.1.1.7	RfSettings	128
6.1.1.7.1	Carrier<Carrier>	129
6.1.1.7.1.1	Frequency	130
6.1.1.7.2	Dcarrier	131
6.1.1.8	Tpc	131
6.1.1.8.1	Ctfc	134
6.1.1.8.1.1	Mlength	134
6.1.1.8.2	Dhib	135
6.1.1.8.3	IlpControl	136

	6.1.1.8.3.1	Tsef	137
	6.1.1.8.3.2	Tsgh	138
	6.1.1.8.4	Limit	139
	6.1.1.8.4.1	Ctfc	140
	6.1.1.8.4.2	IlpControl	141
	6.1.1.8.4.3	EpStep	141
	6.1.1.8.4.4	MaxPower	142
	6.1.1.8.4.5	UserDefined	144
	6.1.1.8.4.6	MinPower	145
	6.1.1.8.4.7	PsGroup	146
	6.1.1.8.4.8	Pstep	147
	6.1.1.8.4.9	Mpedch	148
	6.1.1.8.4.10	UlcM	149
	6.1.1.8.4.11	Pa	149
	6.1.1.8.4.12	Pb	150
	6.1.1.8.5	Monitor	151
	6.1.1.8.6	Mpedch	151
	6.1.1.8.7	UlcM	152
	6.1.1.9	UeChannels	153
	6.1.1.9.1	Carrier<Carrier>	154
	6.1.1.9.1.1	Dpcch	154
	6.1.1.9.1.2	Dpdch	155
	6.1.1.9.1.3	Edpcch	156
	6.1.1.9.1.4	Edpdch<EdpdChannel>	157
	6.1.1.9.1.5	Hsdpcch	159
	6.1.1.9.1.6	Config	160
	6.1.1.10	UeSignal	161
	6.1.1.10.1	Carrier<Carrier>	163
	6.1.1.10.1.1	Scode	163
6.2	Route		164
6.2.1	WcdmaMeas		164
6.2.1.1	Scenario		165
6.2.1.1.1	MaProtocol		166
6.2.1.1.2	Salone		166
6.3	Trigger		167
6.3.1	WcdmaMeas		167
6.3.1.1	MultiEval		167
6.3.1.1.1	Catalog		171
6.3.1.1.2	ListPy		171
6.3.1.2	OlpControl		172
6.3.1.2.1	Catalog		174
6.3.1.3	OoSync		175
6.3.1.3.1	Catalog		177
6.3.1.4	Prach		178
6.3.1.4.1	Catalog		180
6.3.1.5	Tpc		181
6.3.1.5.1	Catalog		184
6.4	WcdmaMeas		184
6.4.1	MultiEval		184
6.4.1.1	Ber		186
6.4.1.2	Carrier<Carrier>		187
6.4.1.2.1	CdError		187
6.4.1.2.1.1	Average		187
6.4.1.2.1.2	Current		188

6.4.1.2.1.3	Maximum	190
6.4.1.2.1.4	StandardDev	191
6.4.1.2.2	CdPower	192
6.4.1.2.2.1	Average	192
6.4.1.2.2.2	Current	193
6.4.1.2.2.3	Maximum	194
6.4.1.2.2.4	Minimum	195
6.4.1.2.2.5	StandardDev	197
6.4.1.2.3	Modulation	198
6.4.1.2.3.1	Average	198
6.4.1.2.3.2	Current	200
6.4.1.2.3.3	Maximum	202
6.4.1.2.3.4	StandardDev	204
6.4.1.2.4	RcdError	207
6.4.1.2.4.1	Average	207
6.4.1.2.4.2	Current	209
6.4.1.2.4.3	Maximum	211
6.4.1.2.4.4	OcInfo	213
6.4.1.2.4.5	Sf	214
6.4.1.2.4.6	StandardDev	215
6.4.1.2.5	Trace	217
6.4.1.2.5.1	CdError	217
6.4.1.2.5.2	Dpcch	218
6.4.1.2.5.3	Average	218
6.4.1.2.5.4	Current	219
6.4.1.2.5.5	Maximum	220
6.4.1.2.5.6	StandardDev	221
6.4.1.2.5.7	Dpdch	222
6.4.1.2.5.8	Average	222
6.4.1.2.5.9	Current	223
6.4.1.2.5.10	Maximum	224
6.4.1.2.5.11	StandardDev	225
6.4.1.2.5.12	Edpcch	227
6.4.1.2.5.13	Average	227
6.4.1.2.5.14	Current	228
6.4.1.2.5.15	Maximum	229
6.4.1.2.5.16	StandardDev	230
6.4.1.2.5.17	Edpdch<EdpdChannel>	231
6.4.1.2.5.18	Average	232
6.4.1.2.5.19	Current	233
6.4.1.2.5.20	Maximum	234
6.4.1.2.5.21	StandardDev	235
6.4.1.2.5.22	Hsdpcch	237
6.4.1.2.5.23	Average	237
6.4.1.2.5.24	Current	238
6.4.1.2.5.25	Maximum	239
6.4.1.2.5.26	StandardDev	240
6.4.1.2.5.27	CdPower	241
6.4.1.2.5.28	Dpcch	241
6.4.1.2.5.29	Average	242
6.4.1.2.5.30	Current	243
6.4.1.2.5.31	Maximum	244
6.4.1.2.5.32	Minimum	245
6.4.1.2.5.33	StandardDev	246

6.4.1.2.5.34	Dpdch	247
6.4.1.2.5.35	Average	247
6.4.1.2.5.36	Current	248
6.4.1.2.5.37	Maximum	249
6.4.1.2.5.38	Minimum	250
6.4.1.2.5.39	StandardDev	251
6.4.1.2.5.40	Edpcch	252
6.4.1.2.5.41	Average	252
6.4.1.2.5.42	Current	254
6.4.1.2.5.43	Maximum	255
6.4.1.2.5.44	Minimum	256
6.4.1.2.5.45	StandardDev	257
6.4.1.2.5.46	Edpdch<EdpdChannel>	258
6.4.1.2.5.47	Average	258
6.4.1.2.5.48	Current	259
6.4.1.2.5.49	Maximum	261
6.4.1.2.5.50	Minimum	262
6.4.1.2.5.51	StandardDev	263
6.4.1.2.5.52	Hsdpcch	264
6.4.1.2.5.53	Average	265
6.4.1.2.5.54	Current	266
6.4.1.2.5.55	Maximum	267
6.4.1.2.5.56	Minimum	268
6.4.1.2.5.57	StandardDev	269
6.4.1.2.5.58	EvMagnitude	270
6.4.1.2.5.59	Peak	270
6.4.1.2.5.60	Average	270
6.4.1.2.5.61	Current	272
6.4.1.2.5.62	Maximum	273
6.4.1.2.5.63	Sdeviaton	274
6.4.1.2.5.64	StandardDev	274
6.4.1.2.5.65	Rms	275
6.4.1.2.5.66	Average	275
6.4.1.2.5.67	Current	276
6.4.1.2.5.68	Maximum	277
6.4.1.2.5.69	Sdeviaton	278
6.4.1.2.5.70	StandardDev	279
6.4.1.2.5.71	FreqError	279
6.4.1.2.5.72	Average	280
6.4.1.2.5.73	Current	281
6.4.1.2.5.74	Maximum	282
6.4.1.2.5.75	StandardDev	283
6.4.1.2.5.76	Merror	284
6.4.1.2.5.77	Peak	284
6.4.1.2.5.78	Average	285
6.4.1.2.5.79	Current	286
6.4.1.2.5.80	Maximum	287
6.4.1.2.5.81	StandardDev	288
6.4.1.2.5.82	Rms	289
6.4.1.2.5.83	Average	289
6.4.1.2.5.84	Current	290
6.4.1.2.5.85	Maximum	291
6.4.1.2.5.86	StandardDev	292
6.4.1.2.5.87	Perror	294

6.4.1.2.5.88	Peak	294
6.4.1.2.5.89	Average	294
6.4.1.2.5.90	Current	295
6.4.1.2.5.91	Maximum	296
6.4.1.2.5.92	StandardDev	297
6.4.1.2.5.93	Rms	298
6.4.1.2.5.94	Average	298
6.4.1.2.5.95	Current	300
6.4.1.2.5.96	Maximum	300
6.4.1.2.5.97	StandardDev	301
6.4.1.2.5.98	Psteps	302
6.4.1.2.5.99	Average	303
6.4.1.2.5.100	Current	304
6.4.1.2.5.101	Maximum	305
6.4.1.2.5.102	Minimum	306
6.4.1.2.5.103	StandardDev	307
6.4.1.2.5.104	RcdError	308
6.4.1.2.5.105	Dpcch	308
6.4.1.2.5.106	Average	309
6.4.1.2.5.107	Current	310
6.4.1.2.5.108	Maximum	311
6.4.1.2.5.109	StandardDev	312
6.4.1.2.5.110	Dpdch	313
6.4.1.2.5.111	Average	313
6.4.1.2.5.112	Current	314
6.4.1.2.5.113	Maximum	315
6.4.1.2.5.114	StandardDev	316
6.4.1.2.5.115	Edpcch	318
6.4.1.2.5.116	Average	318
6.4.1.2.5.117	Current	319
6.4.1.2.5.118	Maximum	320
6.4.1.2.5.119	StandardDev	321
6.4.1.2.5.120	Edpdch<EdpdChannel>	322
6.4.1.2.5.121	Average	323
6.4.1.2.5.122	Current	324
6.4.1.2.5.123	Maximum	325
6.4.1.2.5.124	StandardDev	326
6.4.1.2.5.125	Hsdpcch	328
6.4.1.2.5.126	Average	328
6.4.1.2.5.127	Current	329
6.4.1.2.5.128	Maximum	330
6.4.1.2.5.129	StandardDev	331
6.4.1.2.5.130	Sf	332
6.4.1.2.5.131	Dpcch	332
6.4.1.2.5.132	Dpdch	333
6.4.1.2.5.133	Edpcch	334
6.4.1.2.5.134	Edpdch<EdpdChannel>	335
6.4.1.2.5.135	Hsdpcch	337
6.4.1.2.5.136	UePower	338
6.4.1.2.5.137	Average	338
6.4.1.2.5.138	Current	339
6.4.1.2.5.139	Maximum	340
6.4.1.2.5.140	Minimum	341
6.4.1.2.5.141	StandardDev	342

6.4.1.3	ListPy	343
6.4.1.3.1	CdError	343
6.4.1.3.1.1	Average	344
6.4.1.3.1.2	Current	345
6.4.1.3.1.3	Dpcch	345
6.4.1.3.1.4	Average	346
6.4.1.3.1.5	Current	346
6.4.1.3.1.6	Maximum	347
6.4.1.3.1.7	StandardDev	347
6.4.1.3.1.8	Dpdch	348
6.4.1.3.1.9	Average	348
6.4.1.3.1.10	Current	348
6.4.1.3.1.11	Maximum	349
6.4.1.3.1.12	StandardDev	349
6.4.1.3.1.13	Edpcch	350
6.4.1.3.1.14	Average	350
6.4.1.3.1.15	Current	350
6.4.1.3.1.16	Maximum	351
6.4.1.3.1.17	StandardDev	351
6.4.1.3.1.18	Edpdch<EdpdChannel>	352
6.4.1.3.1.19	Average	352
6.4.1.3.1.20	Current	353
6.4.1.3.1.21	Maximum	353
6.4.1.3.1.22	StandardDev	354
6.4.1.3.1.23	Hsdpcch	354
6.4.1.3.1.24	Average	354
6.4.1.3.1.25	Current	355
6.4.1.3.1.26	Maximum	355
6.4.1.3.1.27	StandardDev	356
6.4.1.3.1.28	Maximum	356
6.4.1.3.1.29	StandardDev	357
6.4.1.3.2	CdPower	358
6.4.1.3.2.1	Average	358
6.4.1.3.2.2	Current	359
6.4.1.3.2.3	Dpcch	360
6.4.1.3.2.4	Average	360
6.4.1.3.2.5	Current	360
6.4.1.3.2.6	Maximum	361
6.4.1.3.2.7	Minimum	361
6.4.1.3.2.8	StandardDev	362
6.4.1.3.2.9	Dpdch	362
6.4.1.3.2.10	Average	362
6.4.1.3.2.11	Current	363
6.4.1.3.2.12	Maximum	363
6.4.1.3.2.13	Minimum	364
6.4.1.3.2.14	StandardDev	364
6.4.1.3.2.15	Edpcch	365
6.4.1.3.2.16	Average	365
6.4.1.3.2.17	Current	365
6.4.1.3.2.18	Maximum	366
6.4.1.3.2.19	Minimum	366
6.4.1.3.2.20	StandardDev	367
6.4.1.3.2.21	Edpdch<EdpdChannel>	367
6.4.1.3.2.22	Average	367

6.4.1.3.2.23	Current	368
6.4.1.3.2.24	Maximum	369
6.4.1.3.2.25	Minimum	369
6.4.1.3.2.26	StandardDev	370
6.4.1.3.2.27	Hsdpcch	370
6.4.1.3.2.28	Average	370
6.4.1.3.2.29	Current	371
6.4.1.3.2.30	Maximum	371
6.4.1.3.2.31	Minimum	372
6.4.1.3.2.32	StandardDev	372
6.4.1.3.2.33	Maximum	373
6.4.1.3.2.34	Minimum	373
6.4.1.3.2.35	StandardDev	374
6.4.1.3.3	Modulation	375
6.4.1.3.3.1	Average	375
6.4.1.3.3.2	Current	376
6.4.1.3.3.3	Evm	377
6.4.1.3.3.4	Peak	377
6.4.1.3.3.5	Average	378
6.4.1.3.3.6	Current	378
6.4.1.3.3.7	Maximum	379
6.4.1.3.3.8	StandardDev	379
6.4.1.3.3.9	Rms	380
6.4.1.3.3.10	Average	380
6.4.1.3.3.11	Current	380
6.4.1.3.3.12	Maximum	381
6.4.1.3.3.13	StandardDev	381
6.4.1.3.3.14	FreqError	382
6.4.1.3.3.15	Average	382
6.4.1.3.3.16	Current	382
6.4.1.3.3.17	Maximum	383
6.4.1.3.3.18	StandardDev	383
6.4.1.3.3.19	IqImbalance	384
6.4.1.3.3.20	Average	384
6.4.1.3.3.21	Current	384
6.4.1.3.3.22	Maximum	385
6.4.1.3.3.23	StandardDev	385
6.4.1.3.3.24	IqOffset	386
6.4.1.3.3.25	Average	386
6.4.1.3.3.26	Current	386
6.4.1.3.3.27	Maximum	387
6.4.1.3.3.28	StandardDev	387
6.4.1.3.3.29	Maximum	388
6.4.1.3.3.30	Merror	389
6.4.1.3.3.31	Peak	389
6.4.1.3.3.32	Average	389
6.4.1.3.3.33	Current	390
6.4.1.3.3.34	Maximum	390
6.4.1.3.3.35	StandardDev	391
6.4.1.3.3.36	Rms	391
6.4.1.3.3.37	Average	391
6.4.1.3.3.38	Current	392
6.4.1.3.3.39	Maximum	392
6.4.1.3.3.40	StandardDev	393

6.4.1.3.3.41	Perror	393
6.4.1.3.3.42	Peak	393
6.4.1.3.3.43	Average	394
6.4.1.3.3.44	Current	394
6.4.1.3.3.45	Maximum	395
6.4.1.3.3.46	StandardDev	395
6.4.1.3.3.47	Rms	396
6.4.1.3.3.48	Average	396
6.4.1.3.3.49	Current	396
6.4.1.3.3.50	Maximum	397
6.4.1.3.3.51	StandardDev	397
6.4.1.3.3.52	StandardDev	398
6.4.1.3.3.53	TtError	399
6.4.1.3.3.54	Average	399
6.4.1.3.3.55	Current	399
6.4.1.3.3.56	Maximum	400
6.4.1.3.3.57	StandardDev	400
6.4.1.3.3.58	UePower	401
6.4.1.3.3.59	Average	401
6.4.1.3.3.60	Current	401
6.4.1.3.3.61	Maximum	402
6.4.1.3.3.62	StandardDev	402
6.4.1.3.4	Pcde	403
6.4.1.3.4.1	Code	403
6.4.1.3.4.2	Current	403
6.4.1.3.4.3	Maximum	404
6.4.1.3.4.4	Current	404
6.4.1.3.4.5	Error	405
6.4.1.3.4.6	Current	405
6.4.1.3.4.7	Maximum	405
6.4.1.3.4.8	Maximum	406
6.4.1.3.4.9	Phase	407
6.4.1.3.4.10	Current	407
6.4.1.3.4.11	Maximum	407
6.4.1.3.5	Phd	408
6.4.1.3.5.1	Current	408
6.4.1.3.6	Segment<Segment>	409
6.4.1.3.6.1	CdError	409
6.4.1.3.6.2	Average	409
6.4.1.3.6.3	Current	410
6.4.1.3.6.4	Maximum	411
6.4.1.3.6.5	StandardDev	412
6.4.1.3.6.6	CdPower	413
6.4.1.3.6.7	Average	413
6.4.1.3.6.8	Current	414
6.4.1.3.6.9	Maximum	415
6.4.1.3.6.10	Minimum	416
6.4.1.3.6.11	StandardDev	417
6.4.1.3.6.12	Modulation	418
6.4.1.3.6.13	Average	418
6.4.1.3.6.14	Current	419
6.4.1.3.6.15	Maximum	420
6.4.1.3.6.16	StandardDev	421
6.4.1.3.6.17	Pcde	422

6.4.1.3.6.18	Current	422
6.4.1.3.6.19	Maximum	423
6.4.1.3.6.20	Phd	424
6.4.1.3.6.21	Current	424
6.4.1.3.6.22	Spectrum	425
6.4.1.3.6.23	Average	425
6.4.1.3.6.24	Current	426
6.4.1.3.6.25	Maximum	427
6.4.1.3.6.26	UePower	429
6.4.1.3.6.27	Current	429
6.4.1.3.7	Spectrum	430
6.4.1.3.7.1	Aclr	430
6.4.1.3.7.2	M<Minus>	430
6.4.1.3.7.3	Average	431
6.4.1.3.7.4	Current	431
6.4.1.3.7.5	Maximum	432
6.4.1.3.7.6	P<Plus>	433
6.4.1.3.7.7	Average	433
6.4.1.3.7.8	Current	434
6.4.1.3.7.9	Maximum	434
6.4.1.3.7.10	Average	435
6.4.1.3.7.11	Cpower	436
6.4.1.3.7.12	Average	436
6.4.1.3.7.13	Current	437
6.4.1.3.7.14	Maximum	437
6.4.1.3.7.15	Current	438
6.4.1.3.7.16	Emask	439
6.4.1.3.7.17	Ab	439
6.4.1.3.7.18	Average	439
6.4.1.3.7.19	Current	440
6.4.1.3.7.20	Maximum	440
6.4.1.3.7.21	Ba	441
6.4.1.3.7.22	Average	441
6.4.1.3.7.23	Current	442
6.4.1.3.7.24	Maximum	442
6.4.1.3.7.25	Bc	443
6.4.1.3.7.26	Average	443
6.4.1.3.7.27	Current	443
6.4.1.3.7.28	Maximum	444
6.4.1.3.7.29	Cb	444
6.4.1.3.7.30	Average	445
6.4.1.3.7.31	Current	445
6.4.1.3.7.32	Maximum	446
6.4.1.3.7.33	Cd	446
6.4.1.3.7.34	Average	446
6.4.1.3.7.35	Current	447
6.4.1.3.7.36	Maximum	447
6.4.1.3.7.37	Dc	448
6.4.1.3.7.38	Average	448
6.4.1.3.7.39	Current	449
6.4.1.3.7.40	Maximum	449
6.4.1.3.7.41	Ef	450
6.4.1.3.7.42	Average	450
6.4.1.3.7.43	Current	450

6.4.1.3.7.44	Maximum	451
6.4.1.3.7.45	Fe	451
6.4.1.3.7.46	Average	452
6.4.1.3.7.47	Current	452
6.4.1.3.7.48	Maximum	453
6.4.1.3.7.49	Had	453
6.4.1.3.7.50	Average	453
6.4.1.3.7.51	Current	454
6.4.1.3.7.52	Maximum	454
6.4.1.3.7.53	Hda	455
6.4.1.3.7.54	Average	455
6.4.1.3.7.55	Current	456
6.4.1.3.7.56	Maximum	456
6.4.1.3.7.57	Maximum	457
6.4.1.3.7.58	Obw	458
6.4.1.3.7.59	Average	458
6.4.1.3.7.60	Current	459
6.4.1.3.7.61	Maximum	459
6.4.1.3.7.62	UePower	459
6.4.1.3.7.63	Average	460
6.4.1.3.7.64	Current	460
6.4.1.3.7.65	Maximum	461
6.4.1.3.8	Sreliability	461
6.4.1.3.9	UePower	462
6.4.1.3.9.1	Current	462
6.4.1.4	Modulation	462
6.4.1.4.1	PhDhsDpcch	463
6.4.1.4.2	Uephhd	464
6.4.1.5	Pcde	466
6.4.1.5.1	Current	466
6.4.1.5.2	Maximum	467
6.4.1.6	Spectrum	468
6.4.1.6.1	Average	468
6.4.1.6.2	Current	471
6.4.1.6.3	Maximum	474
6.4.1.7	State	477
6.4.1.7.1	All	477
6.4.1.8	Trace	478
6.4.1.8.1	Carrier<Carrier>	478
6.4.1.8.1.1	Perror	479
6.4.1.8.1.2	Rms	479
6.4.1.8.1.3	Current	479
6.4.1.8.2	CdeMonitor	480
6.4.1.8.2.1	Isignal	480
6.4.1.8.2.2	Current	480
6.4.1.8.2.3	Qsignal	481
6.4.1.8.2.4	Current	482
6.4.1.8.3	CdpMonitor	482
6.4.1.8.3.1	Isignal	483
6.4.1.8.3.2	Current	483
6.4.1.8.3.3	Qsignal	484
6.4.1.8.3.4	Current	484
6.4.1.8.4	Emask	485
6.4.1.8.4.1	HkfLeft	485

6.4.1.8.4.2	Average	485
6.4.1.8.4.3	Current	486
6.4.1.8.4.4	Maximum	487
6.4.1.8.4.5	HkfRight	489
6.4.1.8.4.6	Average	489
6.4.1.8.4.7	Current	490
6.4.1.8.4.8	Maximum	491
6.4.1.8.4.9	Kfilter	492
6.4.1.8.4.10	Average	492
6.4.1.8.4.11	Current	493
6.4.1.8.4.12	Maximum	494
6.4.1.8.4.13	MfLeft	495
6.4.1.8.4.14	Average	495
6.4.1.8.4.15	Current	496
6.4.1.8.4.16	Maximum	497
6.4.1.8.4.17	MfRight	498
6.4.1.8.4.18	Average	499
6.4.1.8.4.19	Current	500
6.4.1.8.4.20	Maximum	501
6.4.1.8.5	EvMagnitude	502
6.4.1.8.5.1	Chip	502
6.4.1.8.5.2	Average	502
6.4.1.8.5.3	Current	503
6.4.1.8.5.4	Maximum	504
6.4.1.8.6	Iq	505
6.4.1.8.6.1	Current	505
6.4.1.8.7	Merror	506
6.4.1.8.7.1	Chip	506
6.4.1.8.7.2	Average	506
6.4.1.8.7.3	Current	507
6.4.1.8.7.4	Maximum	508
6.4.1.8.8	Perror	508
6.4.1.8.8.1	Chip	509
6.4.1.8.8.2	Average	509
6.4.1.8.8.3	Current	510
6.4.1.8.8.4	Maximum	510
6.4.1.8.9	Phd	511
6.4.1.8.9.1	Current	511
6.4.2	OlpControl	512
6.4.2.1	Carrier<CARRierExt>	514
6.4.2.1.1	UepPower	515
6.4.2.1.1.1	Rup<RampUpCarrier>	515
6.4.2.2	State	516
6.4.2.2.1	All	517
6.4.3	OoSync	518
6.4.3.1	State	520
6.4.3.1.1	All	521
6.4.4	Prach	521
6.4.4.1	OffPower	523
6.4.4.2	Preamble<Preamble>	524
6.4.4.2.1	Current	524
6.4.4.3	State	526
6.4.4.3.1	All	527
6.4.4.4	Trace	527

6.4.4.4.1	EvMagnitude	528
6.4.4.4.1.1	Chip	528
6.4.4.4.1.2	Current	528
6.4.4.4.1.3	Peak	529
6.4.4.4.1.4	Current	529
6.4.4.4.1.5	Rms	530
6.4.4.4.1.6	Current	530
6.4.4.4.2	FreqError	531
6.4.4.4.2.1	Current	531
6.4.4.4.3	Iq	532
6.4.4.4.3.1	Current	532
6.4.4.4.4	Merror	533
6.4.4.4.4.1	Chip	533
6.4.4.4.4.2	Current	533
6.4.4.4.4.3	Peak	534
6.4.4.4.4.4	Current	534
6.4.4.4.4.5	Rms	535
6.4.4.4.4.6	Current	535
6.4.4.4.5	Perror	536
6.4.4.4.5.1	Chip	536
6.4.4.4.5.2	Current	536
6.4.4.4.5.3	Peak	537
6.4.4.4.5.4	Current	537
6.4.4.4.5.5	Rms	538
6.4.4.4.5.6	Current	538
6.4.4.4.6	Psteps	539
6.4.4.4.6.1	Current	539
6.4.4.4.7	UePower	540
6.4.4.4.7.1	Chip	540
6.4.4.4.7.2	Current	540
6.4.4.4.7.3	Current	541
6.4.5	Tpc	542
6.4.5.1	Carrier<Carrier>	543
6.4.5.1.1	Psteps	544
6.4.5.1.1.1	Average	544
6.4.5.1.1.2	Maximum	547
6.4.5.1.1.3	Minimum	550
6.4.5.1.1.4	Statistics	552
6.4.5.1.2	Trace	554
6.4.5.1.2.1	Psteps	554
6.4.5.1.2.2	Current	554
6.4.5.1.2.3	UePower	556
6.4.5.1.2.4	Current	556
6.4.5.1.3	UePower	557
6.4.5.1.3.1	Average	558
6.4.5.1.3.2	Maximum	559
6.4.5.1.3.3	Minimum	561
6.4.5.1.3.4	Statistics	563
6.4.5.2	Dhib	564
6.4.5.2.1	Average	564
6.4.5.2.2	Maximum	565
6.4.5.2.3	Minimum	566
6.4.5.2.4	Minimumc	567
6.4.5.2.5	Statistics	568

6.4.5.3	State	568
6.4.5.3.1	All	569
6.4.5.4	Total	570
6.4.5.4.1	Trace	570
6.4.5.4.1.1	UePower	570
6.4.5.4.1.2	Current	571
6.4.5.4.2	UePower	571
6.4.5.4.2.1	Average	572
6.4.5.4.2.2	Maximum	572
6.4.5.4.2.3	Minimum	573
6.4.5.4.2.4	Statistics	574
7	RsCMPX_WcdmaMeas Utilities	575
8	RsCMPX_WcdmaMeas Logger	581
9	RsCMPX_WcdmaMeas Events	583
10	Index	585
	Index	587



REVISION HISTORY

1.1 RsCMPX_WcdmaMeas

Rohde & Schwarz CMP180 WCDMA Measurement RsCMPX_WcdmaMeas instrument driver.

Basic Hello-World code:

```
from RsCMPX_WcdmaMeas import *

instr = RsCMPX_WcdmaMeas('TCPIP::192.168.2.101::hislip0')
idn = instr.query('*IDN?')
print('Hello, I am: ' + idn)
```

Supported instruments: CMP180

The package is hosted here: <https://pypi.org/project/RsCMPX-WcdmaMeas/>

Documentation: <https://RsCMPX-WcdmaMeas.readthedocs.io/>

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

1.1.1 Version history

Latest release notes summary: Update for FW 5.0.40

Version 5.0.40

- Update for FW 5.0.40

Version 4.0.186

- Fixed documentation

Version 4.0.185

- First released version for FW 4.0.185

GETTING STARTED

2.1 Introduction



RsCMPX_WcdmaMeas 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('3.7.90.38')
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.set_source(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 (for big files transfer)

- Multithreading session locking - you can use multiple threads talking to one instrument at the same time
- Logging feature tailored for SCPI communication - different for binary and ascii data

2.2 Installation

RsCMPX_WcdmaMeas 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 **Packet 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 RsCMPX_WcdmaMeas`

Option 2 - Installing in Pycharm

- In Pycharm Menu **File->Settings->Project->Project Interpreter** click on the '+' button on the top left (the last PyCharm version)
- Type `RsCMPX_WcdmaMeas` 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 6 step for installing the `RsCMPX_WcdmaMeas` offline:

- Download this python script (**Save target as**): [rsinstrument_offline_install.py](#) This installs all the preconditions that the `RsCMPX_WcdmaMeas` needs.
- Execute the script in your offline computer (supported is python 3.6 or newer)
- Download the `RsCMPX_WcdmaMeas` package to your computer from the [pypi.org](https://pypi.org/project/RsCMPX_WcdmaMeas/#files): https://pypi.org/project/RsCMPX_WcdmaMeas/#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\RsCMPX_WcdmaMeas-5.0.40.3.tar`

2.3 Finding Available Instruments

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

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

from RsCMPX_WcdmaMeas import *

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

2.4 Initiating Instrument Session

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

```

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

- Installed RsCMPX_WcdmaMeas Python module Version 5.0.40 or newer from pypi.org
- Installed VISA, for example R&S Visa 5.12 or newer
"""

from RsCMPX_WcdmaMeas import *

# A good practice is to assure that you have a certain minimum version installed
RsCMPX_WcdmaMeas.assert_minimum_version('5.0.40')
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 = RsCMPX_WcdmaMeas(resource_string_1)

idn = driver.utilities.query_str('*IDN?')
print(f"\nHello, I am: '{idn}'")
print(f'RsCMPX_WcdmaMeas 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 2023.

Do not care about specialty of each session kind; RsCMPX_WcdmaMeas 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 = RsCMPX_WcdmaMeas('TCPIP::192.168.56.101::hislip0', id_query=True, reset=True)
```

- Setting `id_query` to `True` (default is `True`) checks, whether your instrument can be used with the `RsCMPX_WcdmaMeas` 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 `RsCMPX_WcdmaMeas` allows you to choose which VISA to use:

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

from RsCMPX_WcdmaMeas import *

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

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

from RsCMPX_WcdmaMeas import *

driver = RsCMPX_WcdmaMeas('TCPIP::192.168.56.101::5025::SOCKET', True, True, "SelectVisa=
↪ 'socket'")
print(f'Visa manufacturer: {driver.utilities.visa_manufacturer}')
```

(continues on next page)

(continued from previous page)

```
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 = RsCMPX_WcdmaMeas('TCPIP::192.168.56.101::hislip0', True, True, "Simulate=True")
```

More option_string tokens are separated by comma:

```
driver = RsCMPX_WcdmaMeas('TCPIP::192.168.56.101::hislip0', 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 RsCMPX_WcdmaMeas objects:

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

from RsCMPX_WcdmaMeas import *

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

2.5 Plain SCPI Communication

After you have opened the session, you can use the instrument-specific part described in the RsCMPX_WcdmaMeas 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 RsCMPX_WcdmaMeas import *

driver = RsCMPX_WcdmaMeas('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 RsCMPX_WcdmaMeas import *

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

driver = RsCMPX_WcdmaMeas('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
```

2.6 Error Checking

RsCMPX_WcdmaMeas 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.

2.7 Exception Handling

The base class for all the exceptions raised by the RsCMPX_WcdmaMeas 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 RsCMPX_WcdmaMeas 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 = RsCMPX_WcdmaMeas('TCPIP::10.112.1.179::hislip0')
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 RsCMPX_WcdmaMeas exceptions
    print(e.args[0])
    print('Some other RsCMPX_WcdmaMeas 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.

2.8 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 RsCMPX_WcdmaMeas, 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 RsCMPX_WcdmaMeas 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')
```

2.9 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")
```

2.10 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 RsCMPX_WcdmaMeas has this covered. And, this feature is quite universal - not just for big files transfer, but for any data in both directions.

RsCMPX_WcdmaMeas 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 RsCMPX_WcdmaMeas 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 = RsCMPX_WcdmaMeas('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 `RsCMPX_WcdmaMeas` 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
```

2.11 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, `RsCMPX_WcdmaMeas` 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 RsCMPX_WcdmaMeas object
"""

import threading
from RsCMPX_WcdmaMeas import *
```

(continues on next page)

(continued from previous page)

```

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

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

import threading
from RsCMPX_WcdmaMeas import *

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

Run the following example:

```

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

import threading
from RsCMPX_WcdmaMeas import *

def execute(session: RsCMPX_WcdmaMeas, 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 = RsCMPX_WcdmaMeas('TCPIP::192.168.56.101::INSTR')
driver2 = RsCMPX_WcdmaMeas('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.

2.12 Logging

Yes, the logging again. This one is tailored for instrument communication. You will appreciate such handy feature when you troubleshoot your program, or just want to protocol the SCPI communication for your test reports.

What can you actually do with the logger?

- Write SCPI communication to a stream-like object, for example console or file, or both simultaneously
- Log only errors and skip problem-free parts; this way you avoid going through thousands lines of texts
- Investigate duration of certain operations to optimize your program's performance
- Log custom messages from your program

Let us take this basic example:

```

"""
Basic logging example to the console
"""

from RsCMPX_WcdmaMeas import *

```

(continues on next page)

(continued from previous page)

```

driver = RsCMPX_WcdmaMeas('TCPIP::192.168.1.101::INSTR')

# Switch ON logging to the console.
driver.utilities.logger.log_to_console = True
driver.utilities.logger.mode = LoggingMode.On
driver.utilities.reset()

# Close the session
driver.close()

```

Console output:

10:29:10.819	TCPIP::192.168.1.101::INSTR	0.976 ms	Write: *RST
10:29:10.819	TCPIP::192.168.1.101::INSTR	1884.985 ms	Status check: OK
10:29:12.704	TCPIP::192.168.1.101::INSTR	0.983 ms	Query OPC: 1
10:29:12.705	TCPIP::192.168.1.101::INSTR	2.892 ms	Clear status: OK
10:29:12.708	TCPIP::192.168.1.101::INSTR	3.905 ms	Status check: OK
10:29:12.712	TCPIP::192.168.1.101::INSTR	1.952 ms	Close: Closing session

The columns of the log are aligned for better reading. Columns meaning:

- (1) Start time of the operation
- (2) Device resource name (you can set an alias)
- (3) Duration of the operation
- (4) Log entry

Tip: You can customize the logging format with `set_format_string()`, and set the maximum log entry length with the properties:

- `abbreviated_max_len_ascii`
- `abbreviated_max_len_bin`
- `abbreviated_max_len_list`

See the full logger help [here](#).

Notice the SCPI communication starts from the line `driver.utilities.reset()`. If you want to log the initialization of the session as well, you have to switch the logging ON already in the constructor:

```
driver = RsCMPX_WcdmaMeas('TCPIP::192.168.56.101::hislip0', options='LoggingMode=On')
```

Parallel to the console logging, you can log to a general stream. Do not fear the programmer's jargon... under the term **stream** you can just imagine a file. To be a little more technical, a stream in Python is any object that has two methods: `write()` and `flush()`. This example opens a file and sets it as logging target:

```

"""
Example of logging to a file
"""

from RsCMPX_WcdmaMeas import *

driver = RsCMPX_WcdmaMeas('TCPIP::192.168.1.101::INSTR')

```

(continues on next page)

(continued from previous page)

```

# We also want to log to the console.
driver.utilities.logger.log_to_console = True

# Logging target is our file
file = open(r'c:\temp\my_file.txt', 'w')
driver.utilities.logger.set_logging_target(file)
driver.utilities.logger.mode = LoggingMode.On

# Instead of the 'TCPIP::192.168.1.101::INSTR', show 'MyDevice'
driver.utilities.logger.device_name = 'MyDevice'

# Custom user entry
driver.utilities.logger.info_raw('----- This is my custom log entry. ---- ')

driver.utilities.reset()

# Close the session
driver.close()

# Close the log file
file.close()

```

Tip: To make the log more compact, you can skip all the lines with Status check: OK:

```
driver.utilities.logger.log_status_check_ok = False
```

Hint: You can share the logging file between multiple sessions. In such case, remember to close the file only after you have stopped logging in all your sessions, otherwise you get a log write error.

For logging to a UDP port in addition to other log targets, use one of the lines:

```
driver.utilities.logger.log_to_udp = True
driver.utilities.logger.log_to_console_and_udp = True
```

You can select the UDP port to log to, the default is 49200:

```
driver.utilities.logger.udp_port = 49200
```

Another cool feature is logging only errors. To make this mode usefull for troubleshooting, you also want to see the circumstances which lead to the errors. Each driver elementary operation, for example, `write_str()`, can generate a group of log entries - let us call them **Segment**. In the logging mode **Errors**, a whole segment is logged only if at least one entry of the segment is an error.

The script below demonstrates this feature. We use a direct SCPI communication to send a misspelled SCPI command *CLS, which leads to instrument status error:

```

"""
Logging example to the console with only errors logged
"""

```

(continues on next page)

(continued from previous page)

```
from RsCMPX_WcdmaMeas import *

driver = RsCMPX_WcdmaMeas('TCPIP::192.168.1.101::INSTR', options='LoggingMode=Errors')

# Switch ON logging to the console.
driver.utilities.logger.log_to_console = True

# Reset will not be logged, since no error occurred there
driver.utilities.reset()

# Now a misspelled command.
driver.utilities.write('*CLaS')

# A good command again, no logging here
idn = driver.utilities.query('*IDN?')

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

Console output:

```
12:11:02.879 TCPIP::192.168.1.101::INSTR    0.976 ms  Write string: *CLaS
12:11:02.879 TCPIP::192.168.1.101::INSTR    6.833 ms  Status check: StatusException:
                                         Instrument error detected: Undefined header;
→ *CLaS
```

Notice the following:

- Although the operation **Write string: *CLaS** finished without an error, it is still logged, because it provides the context for the actual error which occurred during the status checking right after.
- No other log entries are present, including the session initialization and close, because they were all error-free.

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 (163x):
I11I | I13I | I15I | I17I | I21I | I23I | I25I | I27I
I31I | I33I | I35I | I37I | I41I | I43I | I45I | I47I
IFI1 | IFI2 | IFI3 | IFI4 | IFI5 | IFI6 | IQ1I | IQ3I
IQ5I | IQ7I | R10D | R11 | R11C | R11D | R12 | R12C
R12D | R12I | R13 | R13C | R14 | R14C | R14I | R15
```

(continues on next page)

(continued from previous page)

R16	R17	R18	R21	R21C	R22	R22C	R22I
R23	R23C	R24	R24C	R24I	R25	R26	R27
R28	R31	R31C	R32	R32C	R32I	R33	R33C
R34	R34C	R34I	R35	R36	R37	R38	R41
R41C	R42	R42C	R42I	R43	R43C	R44	R44C
R44I	R45	R46	R47	R48	RA1	RA2	RA3
RA4	RA5	RA6	RA7	RA8	RB1	RB2	RB3
RB4	RB5	RB6	RB7	RB8	RC1	RC2	RC3
RC4	RC5	RC6	RC7	RC8	RD1	RD2	RD3
RD4	RD5	RD6	RD7	RD8	RE1	RE2	RE3
RE4	RE5	RE6	RE7	RE8	RF1	RF1C	RF2
RF2C	RF2I	RF3	RF3C	RF4	RF4C	RF4I	RF5
RF5C	RF6	RF6C	RF7	RF7C	RF8	RF8C	RF9C
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 TargetMainState

```
# Example value:  
value = enums.TargetMainState.OFF  
# All values (3x):  
OFF | RDY | RUN
```

3.32 TargetSyncState

```
# Example value:  
value = enums.TargetSyncState.ADJusted  
# All values (2x):  
ADJusted | PENDing
```

3.33 TestCase

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

3.34 TestScenarioB

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

3.35 Type

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

3.36 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('3.7.90.38')
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:
```

(continues on next page)

(continued from previous page)

```
↪ {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.set_source(enums.SourceIntExt.INTERNAL)

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

RSCMPX_WCDMAMEAS API STRUCTURE

Global RepCaps

```
driver = RsCMPX_WcdmaMeas('TCPIP::192.168.2.101::hislip0')
# Instance range: Inst1 .. Inst32
rc = driver.repcap_instance_get()
driver.repcap_instance_set(repcap.Instance.Inst1)
```

class RsCMPX_WcdmaMeas(*resource_name: str, id_query: bool = True, reset: bool = False, options: str = None, direct_session: object = None*)

845 total commands, 4 Subgroups, 0 group commands

Initializes new RsCMPX_WcdmaMeas 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=ivi** - forces usage of National Instruments Visa
- **QueryInstrumentStatus = False** - same as **driver.utilities.instrument_status_checking = False**. Default: **True**
- **WriteDelay = 20, ReadDelay = 5** - Introduces delay of 20ms before each write and 5ms before each read. Default: **0ms** for both
- **OpcWaitMode = OpcQuery** - mode for all the opc-synchronised write/reads. Other modes: **StbPolling, StbPollingSlow, StbPollingSuperSlow**. Default: **StbPolling**
- **AddTermCharToWriteBinBlock = True** - Adds one additional LF to the end of the binary data (some instruments require that). Default: **False**
- **AssureWriteWithTermChar = True** - Makes sure each command/query is terminated with termination character. Default: Interface dependent
- **TerminationCharacter = "\r"** - Sets the termination character for reading. Default: **\n** (LineFeed or LF)
- **DataChunkSize = 10E3** - Maximum size of one write/read segment. If transferred data is bigger, it is split to more segments. Default: **1E6** bytes
- **OpcTimeout = 10000** - same as **driver.utilities.opc_timeout = 10000**. Default: **30000ms**
- **VisaTimeout = 5000** - same as **driver.utilities.visa_timeout = 5000**. Default: **10000ms**

- `ViClearExeMode` = Disabled - `viClear()` execution mode. Default: `execute_on_all`
- `OpcQueryAfterWrite` = True - same as `driver.utilities.opc_query_after_write` = True. Default: False
- `StbInErrorCheck` = False - if true, the driver checks errors with `*STB?` If false, it uses `SYST:ERR?`. Default: True
- `ScpiQuotes` = double'. - for SCPI commands, you can define how strings are quoted. With single or double quotes. Possible values: `single` | `double` | `{char}`. Default: ```single`
- `LoggingMode` = On - Sets the logging status right from the start. Default: Off
- `LoggingName` = 'MyDevice' - Sets the name to represent the session in the log entries. Default: 'resource_name'
- `LogToGlobalTarget` = True - Sets the logging target to the class-property previously set with `RsCMPX_WcdmaMeas.set_global_logging_target()` Default: False
- `LoggingToConsole` = True - Immediately starts logging to the console. Default: False
- `LoggingToUdp` = True - Immediately starts logging to the UDP port. Default: False
- `LoggingUdpPort` = 49200 - UDP port to log to. Default: 49200

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 subsystem.
- **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.

classmethod `clear_global_logging_relative_timestamp() → None`

Clears the global relative timestamp. After this, all the instances using the global relative timestamp continue logging with the absolute timestamps.

close() → None

Closes the active `RsCMPX_WcdmaMeas` session.

classmethod `from_existing_session(session: object, options: str = None) → RsCMPX_WcdmaMeas`

Creates a new `RsCMPX_WcdmaMeas` object with the entered 'session' reused.

Parameters

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

classmethod `get_global_logging_relative_timestamp() → datetime`

Returns global common relative timestamp for log entries.

classmethod `get_global_logging_target()`

Returns global common target stream.

get_session_handle() → object

Returns the underlying session handle.

get_total_execution_time() → timedelta

Returns total time spent by the library on communicating with the instrument. This time is always shorter than `get_total_time()`, since it does not include gaps between the communication. You can reset this counter with `reset_time_statistics()`.

get_total_time() → timedelta

Returns total time spent by the library on communicating with the instrument. This time is always shorter than `get_total_time()`, since it does not include gaps between the communication. You can reset this counter with `reset_time_statistics()`.

static `list_resources(expression: str = '?*::INSTR', visa_select: 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: `'@ivi'`, `'@rs'`

reset_time_statistics() → None

Resets all execution and total time counters. Affects the results of `get_total_time()` and `get_total_execution_time()`

restore_all_repcaps_to_default() → None

Sets all the Group and Global repcaps to their initial values

classmethod `set_global_logging_relative_timestamp(timestamp: datetime)` → None

Sets global common relative timestamp for log entries. To use it, call the following:
`io.utilities.logger.set_relative_timestamp_global()`

classmethod `set_global_logging_relative_timestamp_now()` → None

Sets global common relative timestamp for log entries to this moment. To use it, call the following:
`io.utilities.logger.set_relative_timestamp_global()`.

classmethod `set_global_logging_target(target)` → None

Sets global common target stream that each instance can use. To use it, call the following:
`io.utilities.logger.set_logging_target_global()`. If an instance uses global logging target, it automatically uses the global relative timestamp (if set). You can set the target to None to invalidate it.

Subgroups

6.1 Configure

class ConfigureCls

Configure commands group definition. 164 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1 WcdmaMeas

class WcdmaMeasCls

WcdmaMeas commands group definition. 164 total commands, 10 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1.1 Carrier<Carrier>

RepCap Settings

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

class CarrierCls

Carrier commands group definition. 1 total commands, 1 Subgroups, 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.wcdmaMeas.carrier.clone()
```

Subgroups

6.1.1.1.1 Band

SCPI Command :

```
CONFIgure:WCDMa:MEASurement<instance>:CARRier<carrier>:BAND
```

class BandCls

Band commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(*carrier=Carrier.Default*) → Band

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

Selects the operating band (OB) .

param carrier

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

return

band: OB1, ..., OB22: operating band I 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

set(*band: Band, carrier=Carrier.Default*) → None

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

Selects the operating band (OB) .

param band

OB1, ..., OB22: operating band I 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

param carrier

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

6.1.1.2 Cell

class CellCls

Cell commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1.2.1 Carrier<Carrier>

RepCap Settings

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

class CarrierCls

Carrier commands group definition. 1 total commands, 1 Subgroups, 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.wcdmaMeas.cell.carrier.clone()
```

Subgroups

6.1.1.2.1.1 Scode

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:CELL:CARRier<carrier>:SCODE
```

class ScodeCls

Scode commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(carrier=Carrier.Default) → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:CELL:CARRier<carrier>:SCODE
value: float = driver.configure.wcdmaMeas.cell.carrier.scode.get(carrier = ↵
↵repcap.Carrier.Default)
```

Specifies index i for calculation of the primary downlink scrambling code number by multiplication with 16.

param carrier

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

return

code: No help available

set(code: float, carrier=Carrier.Default) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:CELL:CARRier<carrier>:SCODE
driver.configure.wcdmaMeas.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.

param code

No help available

param carrier

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

6.1.1.3 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 MultiEvalCls

MultiEval commands group definition. 75 total commands, 11 Subgroups, 7 group commands

get_mo_exception() → bool

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:MOEXception
value: bool = driver.configure.wcdmaMeas.multiEval.get_mo_exception()
```

Specifies whether measurement results that the CMP180 identifies as faulty or inaccurate are rejected.

return

meas_on_exception: 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.wcdmaMeas.multiEval.get_ms_count()
```

Selects the total number of measured slots.

return

slot_count: No help available

get_pslot() → int

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

Selects the slot where the CMP180 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 RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.msCount) .

return

slot_number: No help available

get_repetition() → Repeat

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:REPetition
value: enums.Repeat = driver.configure.wcdmaMeas.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: Single-shot measurement CONTinuous: Continuous measurement

get_scondition() → StopCondition

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCONdition
value: enums.StopCondition = driver.configure.wcdmaMeas.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: Continue measurement irrespective of the limit check SLFail: Stop measurement on limit failure

get_synch() → SlotNumber

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

Selects a slot number within the UL WCDMA frames (0 to 14) that the CMP180 displays as the first slot in the measurement interval.

return

slot_number: 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.wcdmaMeas.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. 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: No help available

set_mo_exception(*meas_on_exception: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:MOEXception
driver.configure.wcdmaMeas.multiEval.set_mo_exception(meas_on_exception = False)
```

Specifies whether measurement results that the CMP180 identifies as faulty or inaccurate are rejected.

param meas_on_exception

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.wcdmaMeas.multiEval.set_ms_count(slot_count = 1)
```

Selects the total number of measured slots.

param slot_count

No help available

set_pslot(*slot_number: int*) → None

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

Selects the slot where the CMP180 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 RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.msCount) .

param slot_number

No help available

set_repetition(*repetition: Repeat*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:REPetition
driver.configure.wcdmaMeas.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>:::SCOut to determine the number of measurement intervals per single shot.

param repetition

SINGleshot: Single-shot measurement CONTInuous: Continuous measurement

set_scondition(*stop_condition: StopCondition*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:SCONdition
driver.configure.wcdmaMeas.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: Continue measurement irrespective of the limit check SLFail: Stop measurement on limit failure

set_sync(slot_number: SlotNumber) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:SYNCh
driver.configure.wcdmaMeas.multiEval.set_sync(slot_number = enums.SlotNumber.
↳ANY)
```

Selects a slot number within the UL WCDMA frames (0 to 14) that the CMP180 displays as the first slot in the measurement interval.

param slot_number

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.wcdmaMeas.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. 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

No help available

Cloning the Group

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

Subgroups

6.1.1.3.1 Amode

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:AMODE:MODulation
```

class AmodeCls

Amode commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_modulation() → AnalysisMode

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:AMODE:MODulation
value: enums.AnalysisMode = driver.configure.wcdmaMeas.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: With origin offset NOOFset: No origin offset

set_modulation(analysis_mode: AnalysisMode) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:AMODE:MODulation
driver.configure.wcdmaMeas.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: With origin offset NOOFset: No origin offset

6.1.1.3.2 CdThreshold

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:CDThreshold:MODulation
```

class CdThresholdCls

CdThreshold commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_modulation() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:CDThreshold:MODulation
value: float = driver.configure.wcdmaMeas.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: No help available

set_modulation(threshold: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:CDThreshold:MODulation
driver.configure.wcdmaMeas.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

No help available

6.1.1.3.3 Dmode

SCPI Command :

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

class DmodeCls

Dmode commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_modulation() → DetectionMode

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

Selects the detection mode for uplink WCDMA signals.

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

set_modulation(detection_mode: DetectionMode) → None

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

Selects the detection mode for uplink WCDMA signals.

```
param detection_mode
    A3G: 3GPP Signal Auto
```

6.1.1.3.4 DsFactor

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEValuation:DSFactor:MODulation
```

class DsFactorCls

DsFactor commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_modulation() → SpreadingFactorA

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:DSFactor:MODulation
value: enums.SpreadingFactorA = driver.configure.wcdmaMeas.multiEval.dsFactor.
↳get_modulation()
```

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

```
return
    spreading_factor: Spreading factor 4 to 256
```

set_modulation(spreading_factor: SpreadingFactorA) → None

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

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

param spreading_factor
Spreading factor 4 to 256

6.1.1.3.5 Limit

SCPI Commands :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:IQOffset
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:IQIMbalance
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:CFERror
```

class LimitCls

Limit commands group definition. 23 total commands, 9 Subgroups, 3 group commands

get_cf_error() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:CFERror
value: float or bool = driver.configure.wcdmaMeas.multiEval.limit.get_cf_error()
```

Defines an upper limit for the carrier frequency error.

return
frequency_error: (float or boolean) No help available

get_iq_imbalance() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:IQIMbalance
value: float or bool = driver.configure.wcdmaMeas.multiEval.limit.get_iq_
↳ imbalance()
```

Defines an upper limit for the I/Q imbalance.

return
iq_imbalance: (float or boolean) No help available

get_iq_offset() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:IQOffset
value: float or bool = driver.configure.wcdmaMeas.multiEval.limit.get_iq_
↳ offset()
```

Defines an upper limit for the I/Q origin offset.

return
iq_offset: (float or boolean) No help available

set_cf_error(frequency_error: float) → None

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

Defines an upper limit for the carrier frequency error.

param frequency_error
(float or boolean) No help available

set_iq_imbalance(iq_imbalance: float) → None

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

Defines an upper limit for the I/Q imbalance.

param iq_imbalance
(float or boolean) No help available

set_iq_offset(iq_offset: float) → None

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

Defines an upper limit for the I/Q origin offset.

param iq_offset
(float or boolean) No help available

Cloning the Group

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

Subgroups

6.1.1.3.5.1 Aclr

SCPI Command :

```
CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIMit:ACLR:ABSolute
```

class AclrCls

Aclr commands group definition. 2 total commands, 1 Subgroups, 1 group commands

get_absolute() → float

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIMit:ACLR:ABSolute
value: float or bool = driver.configure.wcdmaMeas.multiEval.limit.aclr.get_
↳absolute()
```

It defines an absolute upper limit for the ACLR. If the absolute upper limit is exceeded, relative limits are evaluated (method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Aclr.Relative.set) .

return

limit_3_m_84: (float or boolean) No help available

set_absolute(limit_3_m_84: float) → None

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

It defines an absolute upper limit for the ACLR. If the absolute upper limit is exceeded, relative limits are evaluated (method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Aclr.Relative.set) .

param limit_3_m_84

(float or boolean) No help available

Cloning the Group

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

Subgroups

6.1.1.3.5.2 Relative

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIMit:ACLR:RELative
```

class RelativeCls

Relative commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class RelativeStruct

Response structure. Fields:

- Channel_First: float or bool: For single uplink carrier: ± 5 MHz from the center frequency For dual uplink carrier: ± 7.5 MHz from the center frequency of both carriers
- Channel_Second: float or bool: For single uplink carrier: ± 10 MHz from the center frequency For dual uplink carrier: ± 12.5 MHz from the center frequency of both carriers

get() → RelativeStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIMit:ACLR:RELative
value: RelativeStruct = driver.configure.wcdmaMeas.multiEval.limit.aclr.
↪relative.get()
```

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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Aclr.absolute) .

return

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

set(channel_first: float, channel_second: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:ACLR:RELative
driver.configure.wcdmaMeas.multiEval.limit.aclr.relative.set(channel_first = 1.
↪0, channel_second = 1.0)
```

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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Aclr.absolute).

param channel_first

(float or boolean) For single uplink carrier: ± 5 MHz from the center frequency For dual uplink carrier: ± 7.5 MHz from the center frequency of both carriers

param channel_second

(float or boolean) For single uplink carrier: ± 10 MHz from the center frequency For dual uplink carrier: ± 12.5 MHz from the center frequency of both carriers

6.1.1.3.5.3 Emask

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EMASk:RELative
```

class EmaskCls

Emask commands group definition. 3 total commands, 2 Subgroups, 1 group commands

class RelativeStruct

Structure for setting input parameters. Fields:

- Point_A: float or bool: No parameter help available
- Point_B: float or bool: No parameter help available
- Point_C: float or bool: No parameter help available
- Point_D: float or bool: No parameter help available
- Point_E: float or bool: No parameter help available
- Point_F: float or bool: No parameter help available

get_relative() → RelativeStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EMASk:RELative
value: RelativeStruct = driver.configure.wcdmaMeas.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_relative(value: RelativeStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EMASk:RELative
structure = driver.configure.wcdmaMeas.multiEval.limit.emask.RelativeStruct()
structure.Point_A: float or bool = 1.0
structure.Point_B: float or bool = 1.0
structure.Point_C: float or bool = 1.0
structure.Point_D: float or bool = 1.0
structure.Point_E: float or bool = 1.0
structure.Point_F: float or bool = 1.0
driver.configure.wcdmaMeas.multiEval.limit.emask.set_relative(value = structure)
```

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.wcdmaMeas.multiEval.limit.emask.clone()
```

Subgroups

6.1.1.3.5.4 Absolute

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EMASk:ABSolute
```

class AbsoluteCls

Absolute commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class AbsoluteStruct

Response structure. Fields:

- Limit_G_3_M_84: float or bool: Absolute limit line G referenced to a 3.84 MHz filter.
- Limit_H_1_Mhz: float or bool: Absolute limit line H is referenced to a 1 MHz or 100 kHz filter, depending on the line H mode.
- Limit_H_30_Khz: float or bool: Absolute limit line H referenced to a 30 kHz filter.
- Limit_Hmode: enums.LimitHmode: Line H mode

get() → AbsoluteStruct

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

Defines absolute limits for the spectrum emission curves.

return

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

set(limit_g_3_m_84: float, limit_h_1_mhz: float, limit_h_30_khz: float, limit_hmode: LimitHmode) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EMASk:ABSolute
driver.configure.wcdmaMeas.multiEval.limit.emask.absolute.set(limit_g_3_m_84 =
↪1.0, limit_h_1_mhz = 1.0, limit_h_30_khz = 1.0, limit_hmode = enums.
↪LimitHmode.A)
```

Defines absolute limits for the spectrum emission curves.

param limit_g_3_m_84

(float or boolean) Absolute limit line G referenced to a 3.84 MHz filter.

param limit_h_1_mhz

(float or boolean) Absolute limit line H is referenced to a 1 MHz or 100 kHz filter, depending on the line H mode.

param limit_h_30_khz

(float or boolean) Absolute limit line H referenced to a 30 kHz filter.

param limit_hmode

Line H mode

6.1.1.3.5.5 Dcarrier

class DcarrierCls

Dcarrier commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1.3.5.6 Absolute

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EMASk:DCARrier:ABSolute
```

class AbsoluteCls

Absolute commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class AbsoluteStruct

Response structure. Fields:

- Point_Ij: float or bool: Absolute limit line I-J referenced to a 1 MHz filter.
- Point_Jk: float or bool: Absolute limit line J-K referenced to a 1 MHz filter.
- Point_Kl: float or bool: Absolute limit line K-L referenced to a 1 MHz filter.
- Point_Mn: float or bool: Absolute limit line M-N referenced to a 30 kHz filter.

get() → AbsoluteStruct

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

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(point_ij: float, point_jk: float, point_kl: float, point_mn: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳ :MEValuation:LIMit:EMASk:DCARrier:ABSolute
driver.configure.wcdmaMeas.multiEval.limit.emask.dcarrier.absolute.set(point_ij,
↳ point_jk = 1.0, point_jk = 1.0, point_kl = 1.0, point_mn = 1.0)
```

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

param point_ij

(float or boolean) Absolute limit line I-J referenced to a 1 MHz filter.

param point_jk

(float or boolean) Absolute limit line J-K referenced to a 1 MHz filter.

param point_kl

(float or boolean) Absolute limit line K-L referenced to a 1 MHz filter.

param point_mn

(float or boolean) Absolute limit line M-N referenced to a 30 kHz filter.

6.1.1.3.5.7 EvMagnitude

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EvMagnitude
```

class EvMagnitudeCls

EvMagnitude commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class EvMagnitudeStruct

Response structure. Fields:

- Rms: float or bool: No parameter help available
- Peak: float or bool: No parameter help available

get() → EvMagnitudeStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:EvMagnitude
value: EvMagnitudeStruct = driver.configure.wcdmaMeas.multiEval.limit.
↳ evMagnitude.get()
```

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.

set(rms: float, peak: float) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIMit:EVMagnitude
driver.configure.wcdmaMeas.multiEval.limit.evMagnitude.set(rms = 1.0, peak = 1.0)
```

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

param rms

(float or boolean) No help available

param peak

(float or boolean) No help available

6.1.1.3.5.8 Merror

SCPI Command :

```
CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIMit:MERRor
```

class MerrorCls

Merror commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class MerrorStruct

Response structure. Fields:

- Rms: float or bool: No parameter help available
- Peak: float or bool: No parameter help available

get() → MerrorStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIMit:MERRor
value: MerrorStruct = driver.configure.wcdmaMeas.multiEval.limit.merror.get()
```

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.

set(rms: float, peak: float) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIMit:MERRor
driver.configure.wcdmaMeas.multiEval.limit.merror.set(rms = 1.0, peak = 1.0)
```

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

param rms

(float or boolean) No help available

param peak

(float or boolean) No help available

6.1.1.3.5.9 Pcontrol

class PcontrolCls

Pcontrol commands group definition. 2 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1.3.5.10 EpStep

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PCONtrol:EPSTep
```

class EpStepCls

EpStep commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class EpStepStruct

Response structure. Fields:

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

get() → EpStepStruct

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

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.

set(expected_0_db: float, expected_1_db: float, expected_2_db: float, expected_3_db: float, expected_4_to_7_db: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PCONtrol:EPSTep
driver.configure.wcdmaMeas.multiEval.limit.pcontrol.epStep.set(expected_0_db =
↳ 1.0, expected_1_db = 1.0, expected_2_db = 1.0, expected_3_db = 1.0, expected_
↳ 4_to_7_db = 1.0)
```

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

param expected_0_db
Tolerance value for power step size 0 dB

param expected_1_db
Tolerance value for power step size 1 dB

param expected_2_db
Tolerance value for power step size 2 dB

param expected_3_db
Tolerance value for power step size 3 dB

param expected_4_to_7_db
Tolerance value for power step size 4 dB to 7 dB

6.1.1.3.5.11 Hsdpcch

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:PCONtrol:HSDPcch
```

class HsdpcchCls

Hsdpcch commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class HsdpcchStruct

Response structure. Fields:

- Enable: bool: Disables | enables the limit check.
- Dtx_To_Nack: float: No parameter help available
- Nack_To_Cqi: float: No parameter help available
- Cqi_To_Dtx: float: No parameter help available
- Test_Case: enums.TestCase: T0DB: measurement below maximum UE power with TPC command = 0 dB T1DB: measurement at maximum UE power with TPC command = 1 dB

get() → HsdpcchStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:PCONtrol:HSDPcch
value: HsdpcchStruct = driver.configure.wcdmaMeas.multiEval.limit.pcontrol.
↳ hsdpcch.get()
```

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.

return

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

set(enable: bool, dtx_to_nack: float, nack_to_cqi: float, cqi_to_dtx: float, test_case: TestCase = None) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:PCONtrol:HSDPcch
driver.configure.wcdmaMeas.multiEval.limit.pcontrol.hsdpcch.set(enable = False,
↳ dtx_to_nack = 1.0, nack_to_cqi = 1.0, cqi_to_dtx = 1.0, test_case = enums.
↳ TestCase.T0DB)
```


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.

param enable

Disables | enables the limit check.

param dtx_to_nack

No help available

param nack_to_cqi

No help available

param cqi_to_dtx

No help available

param test_case

T0DB: measurement below maximum UE power with TPC command = 0 dB T1DB: measurement at maximum UE power with TPC command = 1 dB

6.1.1.3.5.12 Perror

SCPI Command :

```
CONFIGure:WCDMA:MEASurement<instance>:MEvaluation:LIMit:PERror
```

class PerrorCls

Error commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PerrorStruct

Response structure. Fields:

- Rms: float or bool: No parameter help available
- Peak: float or bool: No parameter help available

get() → PerrorStruct

```
# SCPI: CONFIGure:WCDMA:MEASurement<instance>:MEvaluation:LIMit:PERror
value: PerrorStruct = driver.configure.wcdmaMeas.multiEval.limit.perror.get()
```

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 value.

return

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

set(rms: float, peak: float) → None

```
# SCPI: CONFIGure:WCDMA:MEASurement<instance>:MEvaluation:LIMit:PERror
driver.configure.wcdmaMeas.multiEval.limit.perror.set(rms = 1.0, peak = 1.0)
```

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 value.

param rms

(float or boolean) No help available

param peak
(float or boolean) No help available

6.1.1.3.5.13 Phd

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:PHD
```

class PhdCls

Phd commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PhdStruct

Response structure. Fields:

- Enable: bool: Disables | enables the limit check.
- Upper: float: No parameter help available
- Dynamic: float: No parameter help available

get() → PhdStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:PHD
value: PhdStruct = driver.configure.wcdmaMeas.multiEval.limit.phd.get()
```

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.

set(enable: bool, upper: float, dynamic: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:PHD
driver.configure.wcdmaMeas.multiEval.limit.phd.set(enable = False, upper = 1.0,
↳dynamic = 1.0)
```

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

param enable

Disables | enables the limit check.

param upper

No help available

param dynamic

No help available

6.1.1.3.5.14 PhsDpcch

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PHSDpcch
```

class PhsDpcchCls

PhsDpcch commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PhsDpcchStruct

Response structure. Fields:

- Enable: bool: Disables | enables the limit check.
- Measure_Point_A: float: No parameter help available
- Measure_Point_B: float: No parameter help available
- Dynamic: float: No parameter help available

get() → PhsDpcchStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PHSDpcch
value: PhsDpcchStruct = driver.configure.wcdmaMeas.multiEval.limit.phsDpcch.
↳get()
```

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(enable: bool, measure_point_a: float, measure_point_b: float, dynamic: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIMit:PHSDpcch
driver.configure.wcdmaMeas.multiEval.limit.phsDpcch.set(enable = False, measure_
↳point_a = 1.0, measure_point_b = 1.0, dynamic = 1.0)
```

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 enable

Disables | enables the limit check.

param measure_point_a

No help available

param measure_point_b

No help available

param dynamic

No help available

6.1.1.3.5.15 RcdError

SCPI Command :

```
CONFIGure:WCDMA:MEASurement<instance>:MEValuation:LIMit:RCDerror:ECDP
```

class RcdErrorCls

RcdError commands group definition. 8 total commands, 1 Subgroups, 1 group commands

class EcdpStruct

Structure for setting input parameters. Fields:

- Threshold_Bpsk_1: float: Lower ECDP threshold for BPSK requirement 1
- Threshold_Bpsk_2: float: Lower ECDP threshold for BPSK requirement 2
- Limit_Bpsk_1: float: RCDE limit for BPSK requirement 1
- Limit_Bpsk_2: float: RCDE limit for BPSK requirement 2 (limit = this value minus ECDP)
- Threshold_4_Pam_1: float: Lower ECDP threshold for 4PAM requirement 1
- Threshold_4_Pam_2: float: Lower ECDP threshold for 4PAM requirement 2
- Limit_4_Pam_1: float: RCDE limit for 4PAM requirement 1
- Limit_4_Pam_2: float: RCDE limit for 4PAM requirement 2 (limit = this value minus ECDP)

get_ecdp() → EcdpStruct

```
# SCPI: CONFIGure:WCDMA:MEASurement<instance>:MEValuation:LIMit:RCDerror:ECDP
value: EcdpStruct = driver.configure.wcdmaMeas.multiEval.limit.rcdError.get_
    ↪ecdp()
```

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

return

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

set_ecdp(value: EcdpStruct) → None

```
# SCPI: CONFIGure:WCDMA:MEASurement<instance>:MEValuation:LIMit:RCDerror:ECDP
structure = driver.configure.wcdmaMeas.multiEval.limit.rcdError.EcdpStruct()
structure.Threshold_Bpsk_1: float = 1.0
structure.Threshold_Bpsk_2: float = 1.0
structure.Limit_Bpsk_1: float = 1.0
structure.Limit_Bpsk_2: float = 1.0
structure.Threshold_4_Pam_1: float = 1.0
structure.Threshold_4_Pam_2: float = 1.0
structure.Limit_4_Pam_1: float = 1.0
structure.Limit_4_Pam_2: float = 1.0
driver.configure.wcdmaMeas.multiEval.limit.rcdError.set_ecdp(value = structure)
```

Defines upper limits for the relative CDE (RCDE) of BPSK and 4PAM modulated channels. For each modulation format, 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.wcdmaMeas.multiEval.limit.rcdError.clone()
```

Subgroups

6.1.1.3.5.16 Eecdp

class EecdpCls

Eecdp commands group definition. 7 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1.3.5.17 Carrier<Carrier>

RepCap Settings

```
# Range: Nr1 .. Nr2
rc = driver.configure.wcdmaMeas.multiEval.limit.rcdError.eecdp.carrier.repcap_carrier_
↪get()
driver.configure.wcdmaMeas.multiEval.limit.rcdError.eecdp.carrier.repcap_carrier_
↪set(repcap.Carrier.Nr1)
```

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>
```

class CarrierCls

Carrier commands group definition. 7 total commands, 5 Subgroups, 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_Hsdpcch: 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_Hsdpcch: 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*) → GetStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↳:MEvaluation:LIMit:RCDError:EECDp:CARRIER<carrier>
value: GetStruct = driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.RcdError.↳EecdP.Carrier.Dpcch.set
- method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.RcdError.

(continues on next page)

(continued from previous page)

```

↪ EecdP.Carrier.Dpdch.set
  - method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.RcdError.
↪ EecdP.Carrier.Hsdpcch.set
  - method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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: DPCCCH, DPDCH, HS-DPCCCH, E-DPCCCH, 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: SetStruct, carrier=Carrier.Default) → None

```

# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↪ :MEValuation:LIMit:RCDerror:EECDp:CARRier<carrier>
structure = driver.configure.wcdmaMeas.multiEval.limit.rcdError.eecdP.carrier.
↪ SetStruct()
structure.Enable_Dpcch: bool = False
structure.Beta_Dpcch: int = 1
structure.Sf_Dpcch: int = 1
structure.Enable_Dpdch: bool = False
structure.Beta_Dpdch: int = 1
structure.Sf_Dpdch: int = 1
structure.Enable_Hs_Dpcch: bool = False
structure.Beta_Hsdpcch: int = 1
structure.Sf_Hs_Dpcch: int = 1
structure.Enable_Edpcch: bool = False
structure.Beta_Edpcch: int = 1
structure.Sfe_Dpcch: int = 1
structure.Enable_Edpdch_1: bool = False
structure.Beta_Edpdch_1: int = 1
structure.Sfe_Dpd_Ch_1: int = 1
structure.Enable_Edpdch_2: bool = False
structure.Beta_Edpdch_2: int = 1
structure.Sfe_Dpd_Ch_2: int = 1
structure.Enable_Edpdch_3: bool = False
structure.Beta_Edpdch_3: int = 1
structure.Sfe_Dpd_Ch_3: int = 1
structure.Enable_Edpdch_4: bool = False
structure.Beta_Edpdch_4: int = 1
structure.Sfe_Dpd_Ch_4: int = 1
driver.configure.wcdmaMeas.multiEval.limit.rcdError.eecdP.carrier.set(structure,
↪ carrier = repcap.Carrier.Default)

```

(continues on next page)

(continued from previous page)

```

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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.RcdError.
↪ EecdP.Carrier.Dpcch.set
- method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.RcdError.
↪ EecdP.Carrier.Dpdch.set
- method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.RcdError.
↪ EecdP.Carrier.Hsdpcch.set
- method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 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.wcdmaMeas.multiEval.limit.rcdError.eecdP.carrier.clone()

```

Subgroups

6.1.1.3.5.18 Dpcch

SCPI Command :

```

CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIMit:RCDerror:EECDp:CARRier<carrier>
↪ :DPCCh

```

class DpcchCls

Dpcch commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class GetStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled
- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available
- Nominal_Cdp: float: No parameter help available

- Effective_Cdp: float: No parameter help available

get(*carrier=Carrier.Default*) → GetStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↳ :MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>:DPCCh
value: GetStruct = driver.configure.wcdmaMeas.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 also returns 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*) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↳ :MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>:DPCCh
driver.configure.wcdmaMeas.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 also returns the nominal CDP and effective CDP resulting from these settings.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

param carrier

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

6.1.1.3.5.19 Dpdch

SCPI Command :

```
CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>
↳ :DPDCh
```

class DpdchCls

Dpdch commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class GetStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled

- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available
- Nominal_Cdp: float: No parameter help available
- Effective_Cdp: float: No parameter help available

get(*carrier=Carrier.Default*) → GetStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↳ :MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>:DPDCh
value: GetStruct = driver.configure.wcdmaMeas.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 also returns 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*) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↳ :MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>:DPDCh
driver.configure.wcdmaMeas.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 also returns the nominal CDP and effective CDP resulting from these settings.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

param carrier

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

6.1.1.3.5.20 Edpcch

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIMit:RCDerror:EECDp:CARRier<carrier>
↳:EDPCch
```

class EdpcchCls

Edpcch commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class GetStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled
- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available
- Nominal_Cdp: float: No parameter help available
- Effective_Cdp: float: No parameter help available

get(*carrier=Carrier.Default*) → GetStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:MEValuation:LIMit:RCDerror:EECDp:CARRier<carrier>:EDPCch
value: GetStruct = driver.configure.wcdmaMeas.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 also returns 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*) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳:MEValuation:LIMit:RCDerror:EECDp:CARRier<carrier>:EDPCch
driver.configure.wcdmaMeas.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 also returns the nominal CDP and effective CDP resulting from these settings.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

param carrier

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

6.1.1.3.5.21 Edpdch<EdpdChannel>**RepCap Settings**

```
# Range: Nr1 .. Nr4
rc = driver.configure.wcdmaMeas.multiEval.limit.rcdError.eecdpcarrier.edpdch.repcap_
↪edpdChannel_get()
driver.configure.wcdmaMeas.multiEval.limit.rcdError.eecdpcarrier.edpdch.repcap_
↪edpdChannel_set(repcap.EdpdChannel.Nr1)
```

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>
↪:EDPDch<nr>
```

class EdpdchCls

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

class GetStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled
- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available
- Nominal_Cdp: float: No parameter help available
- Effective_Cdp: float: No parameter help available

get(carrier=Carrier.Default, edpdChannel=EdpdChannel.Default) → GetStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↪:MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>:EDPDch<nr>
value: GetStruct = driver.configure.wcdmaMeas.multiEval.limit.rcdError.eecdpcarrier.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 also returns 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, edpdChannel=EdpdChannel.Default) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↳:MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>:EDPDch<nr>
driver.configure.wcdmaMeas.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 also returns the nominal CDP and effective CDP resulting from these settings.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

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.wcdmaMeas.multiEval.limit.rcdError.eecdP.carrier.edpdch.clone()
```

6.1.1.3.5.22 Hsdpcch

SCPI Command :

```
CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIMit:RCDError:EECDp:CARRier<carrier>
↳:HSDPcch
```

class HsdpcchCls

Hsdpcch commands group definition. 2 total commands, 1 Subgroups, 1 group commands

class GetStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled
- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available

- Nominal_Cdp: float: No parameter help available
- Effective_Cdp: float: No parameter help available

get(*carrier=Carrier.Default*) → GetStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↪:MEvaluation:LIMit:RCDerror:EEDp:CARRier<carrier>:HSDPcch
value: GetStruct = driver.configure.wcdmaMeas.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 also returns 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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*) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↪:MEvaluation:LIMit:RCDerror:EEDp:CARRier<carrier>:HSDPcch
driver.configure.wcdmaMeas.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 also returns 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.RcdError.EecdP.Carrier.Hsdpcch.Config.set.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

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.wcdmaMeas.multiEval.limit.rcdError.eecdP.carrier.hsdpcch.
↳ clone()
```

Subgroups

6.1.1.3.5.23 Config

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>
↳ :HSDPcch:CONFig
```

class ConfigCls

Config commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(carrier=Carrier.Default) → Type

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳ :MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>:HSDPcch:CONFig
value: enums.Type = driver.configure.wcdmaMeas.multiEval.limit.rcdError.eecdP.
↳ 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.

param carrier

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

return

type_py: No help available

set(type_py: Type, carrier=Carrier.Default) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>
↳ :MEvaluation:LIMit:RCDerror:EECDp:CARRier<carrier>:HSDPcch:CONFig
driver.configure.wcdmaMeas.multiEval.limit.rcdError.eecdP.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.

param type_py

No help available

param carrier

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

6.1.1.3.6 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:CMODE
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:NCONNECTIONS
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST

```

class ListPyCls

ListPy commands group definition. 15 total commands, 2 Subgroups, 6 group commands

get_cmode() → ParameterSetMode

```

# SCPI: CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:CMODE
value: enums.ParameterSetMode = driver.configure.wcdmaMeas.multiEval.listPy.get_
    ↪ cmode()

```

Sets the connector mode, selecting whether all list mode segments use the same RF connection.

return

connector_mode: - GLOBAL: Use the same RF connection for all segments, see ROUTe:WCDMa:MEASi:SPATH. - LIST: Assign a connection to each segment, see CONFIGure:WCDMa:MEASi:MEValuation:LIST:SEGMENTno:CIDX.

get_count() → int

```

# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:COUNT
value: int = driver.configure.wcdmaMeas.multiEval.listPy.get_count()

```

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

return

segments: No help available

get_eoffset() → int

```

# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:EOffset
value: int = driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas.Trigger.WcdmaMeas.MultiEval.delay).

return

offset: No help available

get_nconnections() → int

```

# SCPI: CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:NCONNECTIONS
value: int = driver.configure.wcdmaMeas.multiEval.listPy.get_nconnections()

```

Sets the number of connections to be defined for the list mode, for connector mode LIST. Define the connections via ROUTe:WCDMa:MEAS<i>:SPATH.

return

no_of_connections: The maximum number of connections is limited by the number of connectors per smart channel.

get_os_index() → int

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

Enables or disables the list mode.

return

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

set_cmode(connector_mode: ParameterSetMode) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:CMODE
driver.configure.wcdmaMeas.multiEval.listPy.set_cmode(connector_mode = enums.
↳ParameterSetMode.GLOBal)
```

Sets the connector mode, selecting whether all list mode segments use the same RF connection.

param connector_mode

- GLOBal: Use the same RF connection for all segments, see ROUTe:WCDMa:MEASi:SPATH.
- LIST: Assign a connection to each segment, see CONFIGure:WCDMa:MEASi:MEValuation:LIST:SEGMentno:CIDX.

set_count(segments: int) → None

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

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

param segments

No help available

set_eoffset(offset: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:EOffset
driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas.Trigger.WcdmaMeas.MultiEval.delay).

param offset

No help available

set_nconnections(no_of_connections: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:NCONnections
driver.configure.wcdmaMeas.multiEval.listPy.set_nconnections(no_of_connections_
↪= 1)
```

Sets the number of connections to be defined for the list mode, for connector mode LIST. Define the connections via ROUTe:WCDMa:MEAS<i>:SPATH.

param no_of_connections

The maximum number of connections is limited by the number of connectors per smart channel.

set_os_index(index: int) → None

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

Enables or disables the list mode.

param enable

OFF: Disable list mode ON: Enable list mode

Cloning the Group

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

Subgroups**6.1.1.3.6.1 Segment<Segment>****RepCap Settings**

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

class SegmentCls

Segment commands group definition. 8 total commands, 8 Subgroups, 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.wcdmaMeas.multiEval.listPy.segment.clone()
```

Subgroups**6.1.1.3.6.2 CdPower****SCPI Command :**

```
CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:CDPower
```

class CdPowerCls

CdPower commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class CdPowerStruct

Response structure. Fields:

- Mod_Statistics: int: The statistical length is limited by the length of the segment (see [CMDLINKRESOLVED Configure.WcdmaMeas.MultiEval.ListPy.Segment.Setup#set CMDLINKRESOLVED]) .
- Enable_Cdp: bool: OFF: Disable measurement ON: Enable measurement of code domain power.
- Enable_Cde: bool: Disable or enable measurement of code domain error.
- Enable_Pcde: bool: Disable or enable measurement of peak code domain error.

get(segment=Segment.Default) → CdPowerStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:CDPower
value: CdPowerStruct = driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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(mod_statistics: int, enable_cdp: bool, enable_cde: bool, enable_pcde: bool = None, segment=Segment.Default) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:CDPower
driver.configure.wcdmaMeas.multiEval.listPy.segment.cdPower.set(mod_statistics_
↳= 1, enable_cdp = False, enable_cde = False, enable_pcde = False, 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.Segment.Modulation.set) .

param mod_statistics

The statistical length is limited by the length of the segment (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.Segment.Setup.set)

param enable_cdp

OFF: Disable measurement ON: Enable measurement of code domain power.

param enable_cde

Disable or enable measurement of code domain error.

param enable_pcde

Disable or enable measurement of peak code domain error.

param segment

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

6.1.1.3.6.3 Cidx

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>:CIDX
```

class CidxCls

Cidx commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(segment=Segment.Default) → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>:CIDX
value: int = driver.configure.wcdmaMeas.multiEval.listPy.segment.cidX.
↳get(segment = repcap.Segment.Default)
```

Selects the RF connection index for segment <no>. For a definition of the connection indices, see ROUTe:WCDMa:MEAS<i>:SPATH.

param segment

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

return

connection_index: Index of the connection to be used for the segment.

set(connection_index: int, segment=Segment.Default) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<Instance>:MEValuation:LIST:SEGment<nr>:CIDX
driver.configure.wcdmaMeas.multiEval.listPy.segment.cidX.set(connection_index =
↪1, segment = repcap.Segment.Default)
```

Selects the RF connection index for segment <no>. For a definition of the connection indices, see ROUTe:WCDMa:MEAS<i>:SPATh.

param connection_index

Index of the connection to be used for the segment.

param segment

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

6.1.1.3.6.4 Modulation

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:MODulation
```

class ModulationCls

Modulation commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class ModulationStruct

Structure for setting input parameters. Fields:

- Mod_Statistics: int: The statistical length is limited by the length of the segment (see [CMDLINKRESOLVED Configure.WcdmaMeas.MultiEval.ListPy.Segment.Setup#set CMDLINKRESOLVED]) .
- Enable_Ue_Power: bool: OFF: Disable measurement. ON: Enable measurement of UE power.
- Enable_Evm: bool: Disable or enable measurement of EVM.
- Enable_Mag_Error: bool: Disable or enable measurement of magnitude error.
- Enable_Phase_Err: bool: Disable or enable measurement of phase error.
- Enable_Freq_Error: bool: Disable or enable measurement of frequency error.
- Enable_Iq: bool: Disable or enable measurement of I/Q origin offset and imbalance.

get(segment=Segment.Default) → ModulationStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↪:MODulation
value: ModulationStruct = driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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: ModulationStruct, segment=Segment.Default) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:MODulation
structure = driver.configure.wcdmaMeas.multiEval.listPy.segment.modulation.
↳ModulationStruct()
structure.Mod_Statistics: int = 1
structure.Enable_Ue_Power: bool = False
structure.Enable_Evm: bool = False
structure.Enable_Mag_Error: bool = False
structure.Enable_Phase_Err: bool = False
structure.Enable_Freq_Error: bool = False
structure.Enable_Iq: bool = False
driver.configure.wcdmaMeas.multiEval.listPy.segment.modulation.set(structure,
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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’)

6.1.1.3.6.5 Phd

SCPI Command :

```
CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:PHD
```

class PhdCls

Phd commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(segment=Segment.Default) → bool

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:PHD
value: bool = driver.configure.wcdmaMeas.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: Disable measurement ON: Enable measurement of phase discontinuity

set(enable_phd: bool, segment=Segment.Default) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:PHD
driver.configure.wcdmaMeas.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: Disable measurement ON: Enable measurement of phase discontinuity

param segment

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

6.1.1.3.6.6 Setup

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:SETup
```

class SetupCls

Setup commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class SetupStruct

Response structure. Fields:

- Segment_Length: int: 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.
- Level: float: 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 specifications document.
- Frequency: float: No parameter help available
- Retrigger: enums.Rettrigger: 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 [CMDLINKRESOLVED Trigger.WcdmaMeas.MultiEval.ListPy#Mode CMDLINKRESOLVED]) . OFF: measure the segment without retrigger ON: trigger event required, trigger source configured via [CMDLINKRESOLVED Trigger.WcdmaMeas.MultiEval#Source CMDLINKRESOLVED] IFPower: trigger event required, IF Power trigger IFPSync: trigger event required, IF Power (Sync) trigger

get(segment=Segment.Default) → SetupStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:SETup
value: SetupStruct = driver.configure.wcdmaMeas.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(segment_length: int, level: float, frequency: float, retrigger: Retrigger = None, segment=Segment.Default) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:SETup
driver.configure.wcdmaMeas.multiEval.listPy.segment.setup.set(segment_length =
↪ 1, level = 1.0, frequency = 1.0, retrigger = enums.Retrigger.IFPower, 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_length

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.

param level

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 specifications document.

param frequency

No help available

param retrigger

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 method RsCMPX_WcdmaMeas.Trigger.WcdmaMeas.MultiEval.ListPy.mode) . OFF: measure the segment without retrigger ON: trigger event required, trigger source configured via method RsCMPX_WcdmaMeas.Trigger.WcdmaMeas.MultiEval.source IF-Power: trigger event required, IF Power trigger IFPSync: trigger event required, IF Power (Sync) trigger

param segment

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

6.1.1.3.6.7 SingleCmw

class SingleCmwCls

SingleCmw commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1.3.6.8 Connector

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:CMWS:CONNECTor
```

class ConnectorCls

Connector commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(segment=Segment.Default) → CmwsConnector

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

No command help available

param segment

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

return

cmws_connector: No help available

set(cmws_connector: CmwsConnector, segment=Segment.Default) → None

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

No command help available

param cmws_connector

No help available

param segment

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

6.1.1.3.6.9 Spectrum

SCPI Command :

```
CONFigure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:SPECtrum
```

class SpectrumCls

Spectrum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class SpectrumStruct

Response structure. Fields:

- Spec_Statistics: int: The statistical length is limited by the length of the segment (see [CMDLINKRESOLVED Configure.WcdmaMeas.MultiEval.ListPy.Segment.Setup#set CMDLINKRESOLVED]) .
- Enable_Aclr: bool: OFF: Disable measurement. ON: Enable measurement of ACLR.
- Enable_Emask: bool: Disable or enable measurement of spectrum emission mask.
- Enable_Obw: bool: Disable or enable measurement of occupied bandwidth.

get(segment=Segment.Default) → SpectrumStruct

```
# SCPI: CONFigure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:SPECtrum
value: SpectrumStruct = driver.configure.wcdmaMeas.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(spec_statistics: int, enable_aclr: bool, enable_emask: bool, enable_obw: bool, segment=Segment.Default) → None

```
# SCPI: CONFigure:WCDma:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳:SPECtrum
driver.configure.wcdmaMeas.multiEval.listPy.segment.spectrum.set(spec_
↳statistics = 1, enable_aclr = False, enable_emask = False, enable_obw = False,
↳ 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 spec_statistics

The statistical length is limited by the length of the segment (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.Segment.Setup.set)

param enable_aclr

OFF: Disable measurement. ON: Enable measurement of ACLR.

param enable_emask

Disable or enable measurement of spectrum emission mask.

param enable_obw

Disable or enable measurement of occupied bandwidth.

param segment

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

6.1.1.3.6.10 UePower**SCPI Command :**

```
CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:UEPower
```

class UePowerCls

UePower commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(segment=Segment.Default) → bool

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:UEPower
value: bool = driver.configure.wcdmaMeas.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: Disable measurement ON: Enable measurement of UE power

set(enable_ue_power: bool, segment=Segment.Default) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:UEPower
driver.configure.wcdmaMeas.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: Disable measurement ON: Enable measurement of UE power

param segment

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

6.1.1.3.6.11 SingleCmw

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIST:CMWS:CMODE
```

class SingleCmwCls

SingleCmw commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_cmode() → ParameterSetMode

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIST:CMWS:CMODE
value: enums.ParameterSetMode = driver.configure.wcdmaMeas.multiEval.listPy.
↳ singleCmw.get_cmode()
```

No command help available

```
return
connector_mode: No help available
```

set_cmode(connector_mode: ParameterSetMode) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:LIST:CMWS:CMODE
driver.configure.wcdmaMeas.multiEval.listPy.singleCmw.set_cmode(connector_mode,
↳ enums.ParameterSetMode.GLOBal)
```

No command help available

```
param connector_mode
No help available
```

6.1.1.3.7 Mperiod

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEValuation:MPERiod:MODulation
```

class MperiodCls

Mperiod commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_modulation() → MeasPeriod

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:MPERiod:MODulation
value: enums.MeasPeriod = driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount.

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

set_modulation(*meas_period: MeasPeriod*) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:MEValuation:MPERiod:MODulation
driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount.

param meas_period

FULLslot: Full-slot measurement HALFslot: Half-slot measurement

6.1.1.3.8 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
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 ResultCls

Result commands group definition. 20 total commands, 1 Subgroups, 17 group commands

class AllStruct

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

- Enable_Evm: bool: Error vector magnitude OFF: Do not evaluate results. ON: Evaluate results
- Enable_Mag_Error: bool: Magnitude error
- Enable_Phase_Err: bool: Phase error
- Enable_Aclr: bool: Adjacent channel leakage power ratio
- Enable_Emask: bool: Spectrum emission mask
- Enable_Cd_Monitor: bool: Code domain monitor
- Enable_Cdp: bool: Code domain power
- Enable_Cde: bool: Code domain error
- Enable_Evm_Chip: bool: EVM vs chip

- Enable_Merr_Chip: bool: Magnitude error vs chip
- Enable_Ph_Err_Chip: bool: Phase error vs chip
- Enable_Ue_Power: bool: UE power
- Enable_Freq_Error: bool: Frequency error
- Enable_Phase_Disc: bool: Phase discontinuity
- Enable_Pow_Steps: bool: Power steps
- Enable_Ber: bool: Bit error rate
- Enable_Iq: bool: Optional setting parameter. I/Q constellation diagram
- Enable_Rcde: bool: Optional setting parameter. Relative CDE
- Enable_Txm: bool: Optional setting parameter. TX measurement scalar results

get_aclr() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return

enable_aclr: No help available

get_all() → AllStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult[:ALL]
value: AllStruct = driver.configure.wcdmaMeas.multiEval.result.get_all()
```

Enables or disables the evaluation of results 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.wcdmaMeas.multiEval.result.get_ber()
```

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return

enable_ber: No help available

get_cd_error() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return

enable_cde: No help available

get_cd_power() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return

enable_cdp: No help available

get_cdp_monitor() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return

enable_cd_monitor: No help available

get_emask() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return

enable_emask: No help available

get_ev_magnitude() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return
enable_evm: No help available

get_freq_error() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return
enable_freq_error: No help available

get_iq() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return
enable_iq: No help available

get_merror() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return
enable_mag_error: No help available

get_perror() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return
enable_phase_err: No help available

get_phd() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return
enable_phase_disc: No help available

get_psteps() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return
enable_pow_steps: No help available

get_rcd_error() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return
enable_rcde: No help available

get_txm() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return

enable_ue_power: No help available

set_aclr(enable_aclr: bool) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_aclr

OFF: Do not evaluate results. ON: Evaluate the results.

set_all(value: AllStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:RESult[:ALL]
structure = driver.configure.wcdmaMeas.multiEval.result.AllStruct()
structure.Enable_Evm: bool = False
structure.Enable_Mag_Error: bool = False
structure.Enable_Phase_Err: bool = False
structure.Enable_Aclr: bool = False
structure.Enable_Emask: bool = False
structure.Enable_Cd_Monitor: bool = False
structure.Enable_Cdp: bool = False
structure.Enable_Cde: bool = False
structure.Enable_Evm_Chip: bool = False
structure.Enable_Merr_Chip: bool = False
structure.Enable_Ph_Err_Chip: bool = False
structure.Enable_Ue_Power: bool = False
structure.Enable_Freq_Error: bool = False
structure.Enable_Phase_Disc: bool = False
```

(continues on next page)

(continued from previous page)

```

structure.Enable_Pow_Steps: bool = False
structure.Enable_Ber: bool = False
structure.Enable_Iq: bool = False
structure.Enable_Rcde: bool = False
structure.Enable_Txm: bool = False
driver.configure.wcdmaMeas.multiEval.result.set_all(value = structure)

```

Enables or disables the evaluation of results 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.wcdmaMeas.multiEval.result.set_ber(enable_ber = False)

```

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_ber

OFF: Do not evaluate results. ON: Evaluate the results.

set_cd_error(*enable_cde: bool*) → None

```

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

```

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_cde

OFF: Do not evaluate results. ON: Evaluate the results.

set_cd_power(*enable_cdp: bool*) → None

```

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

```

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_cdp

OFF: Do not evaluate results. ON: Evaluate the results.

set_cdp_monitor(*enable_cd_monitor: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_cd_monitor

OFF: Do not evaluate results. ON: Evaluate the results.

set_emask(*enable_emask: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_emask

OFF: Do not evaluate results. ON: Evaluate the results.

set_ev_magnitude(*enable_evm: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_evm

OFF: Do not evaluate results. ON: Evaluate the results.

set_freq_error(*enable_freq_error: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_freq_error

OFF: Do not evaluate results. ON: Evaluate the results.

set_iq(*enable_iq: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_iq

OFF: Do not evaluate results. ON: Evaluate the results.

set_merror(*enable_mag_error: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_mag_error

OFF: Do not evaluate results. ON: Evaluate the results.

set_perror(*enable_phase_err: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_phase_err

OFF: Do not evaluate results. ON: Evaluate the results.

set_phd(*enable_phase_disc: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_phase_disc

OFF: Do not evaluate results. ON: Evaluate the results.

set_psteps(*enable_pow_steps: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_pow_steps

OFF: Do not evaluate results. ON: Evaluate the results.

set_rcd_error(*enable_rcde: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_rcde

OFF: Do not evaluate results. ON: Evaluate the results.

set_txm(*enable_txm: bool*) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_ue_power

OFF: Do not evaluate results. ON: Evaluate the results.

Cloning the Group

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

Subgroups

6.1.1.3.8.1 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 ChipCls

Chip commands group definition. 3 total commands, 0 Subgroups, 3 group commands

get_evm() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

```
return
    enable_evm_chip: No help available
```

get_merror() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

```
return
    enable_merr_chip: No help available
```

get_perror() → bool

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain

error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

return

enable_ph_err_chip: No help available

set_evm(enable_evm_chip: bool) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_evm_chip

OFF: Do not evaluate results. ON: Evaluate the results.

set_merror(enable_merr_chip: bool) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_merr_chip

OFF: Do not evaluate results. ON: Evaluate the results.

set_perror(enable_ph_err_chip: bool) → None

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

Enables or disables the evaluation of results of the multi-evaluation measurement identified by the last command mnemonics: error vector magnitude (EVM) , magnitude error, phase error, adjacent channel leakage power ratio, spectrum emission mask, code domain monitor, code domain power, code domain error (CDE) , EVM vs chip, magnitude error vs chip, phase error vs chip, UE power, frequency error, phase discontinuity, power steps, bit error rate, I/Q constellation, and relative CDE results.

param enable_ph_err_chip

OFF: Do not evaluate results. ON: Evaluate the results.

6.1.1.3.9 Rotation

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEValuation:ROTation:MODulation
```

class RotationCls

Rotation commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_modulation() → int

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

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

return

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

set_modulation(rotation: int) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:ROTation:MODulation
driver.configure.wcdmaMeas.multiEval.rotation.set_modulation(rotation = 1)
```

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

param rotation

The entered value is rounded to 0 deg or 45 deg. 0 deg: constellation points on the I- and Q-axes 45 deg: constellation points on angle bisectors between the I- and Q-axes

6.1.1.3.10 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 ScoutCls

Scout commands group definition. 3 total commands, 0 Subgroups, 3 group commands

get_ber() → int

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:MEValuation:SCount:BER
value: int = driver.configure.wcdmaMeas.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: Number of transport blocks

get_modulation() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCount:MODulation
value: int = driver.configure.wcdmaMeas.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: Number of measurement intervals

get_spectrum() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCount:SPECTrum
value: int = driver.configure.wcdmaMeas.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: Number of measurement intervals

set_ber(statistic_count: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCount:BER
driver.configure.wcdmaMeas.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
Number of transport blocks

set_modulation(statistic_count: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCount:MODulation
driver.configure.wcdmaMeas.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
Number of measurement intervals

set_spectrum(statistic_count: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:MEValuation:SCount:SPECTrum
driver.configure.wcdmaMeas.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
Number of measurement intervals

6.1.1.3.11 Sscalar

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:MEValuation:SScalar:MODulation
```

class SscalarCls

Sscalar commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_modulation() → float

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

Selects a particular slot or half-slot within the measurement length where the CMP180 evaluates the statistical measurement results for multislot measurements. The slot number must be smaller than the number of measured slots (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount) .

return
slot_number: No help available

set_modulation(slot_number: float) → None

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

Selects a particular slot or half-slot within the measurement length where the CMP180 evaluates the statistical measurement results for multislot measurements. The slot number must be smaller than the number of measured slots (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount) .

param slot_number
No help available

6.1.1.4 OlpControl

SCPI Commands :

```
CONFigure:WCDMa:MEASurement<instance>:OLPControl:TOUT
CONFigure:WCDMa:MEASurement<instance>:OLPControl:MOEXception
CONFigure:WCDMa:MEASurement<instance>:OLPControl:LIMit
```

class OlpControlCls

OlpControl commands group definition. 3 total commands, 0 Subgroups, 3 group commands

get_limit() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OLPControl:LIMit
value: float = driver.configure.wcdmaMeas.olpControl.get_limit()
```

No command help available

return
olp_limit: No help available

get_mo_exception() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:MOEXception
value: bool = driver.configure.wcdmaMeas.olpControl.get_mo_exception()
```

No command help available

```
return
    meas_on_exception: No help available
```

get_timeout() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:TOUT
value: float = driver.configure.wcdmaMeas.olpControl.get_timeout()
```

No command help available

```
return
    timeout: No help available
```

set_limit(olp_limit: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:LIMit
driver.configure.wcdmaMeas.olpControl.set_limit(olp_limit = 1.0)
```

No command help available

```
param olp_limit
    No help available
```

set_mo_exception(meas_on_exception: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:MOEXception
driver.configure.wcdmaMeas.olpControl.set_mo_exception(meas_on_exception =
↪False)
```

No command help available

```
param meas_on_exception
    No help available
```

set_timeout(timeout: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OLPControl:TOUT
driver.configure.wcdmaMeas.olpControl.set_timeout(timeout = 1.0)
```

No command help available

```
param timeout
    No help available
```

6.1.1.5 OoSync

SCPI Commands :

```
CONFigure:WCDMa:MEASurement<instance>:OOSync:AADPchlevel
CONFigure:WCDMa:MEASurement<instance>:OOSync:TOUT
```

class OoSyncCls

OoSync commands group definition. 5 total commands, 1 Subgroups, 2 group commands

get_aa_dpch_level() → bool

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:AADPchlevel
value: bool = driver.configure.wcdmaMeas.ooSync.get_aa_dpch_level()
```

No command help available

return
enable: No help available

get_timeout() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:TOUT
value: float = driver.configure.wcdmaMeas.ooSync.get_timeout()
```

No command help available

return
timeout: No help available

set_aa_dpch_level(enable: bool) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:AADPchlevel
driver.configure.wcdmaMeas.ooSync.set_aa_dpch_level(enable = False)
```

No command help available

param enable
No help available

set_timeout(timeout: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:OOSync:TOUT
driver.configure.wcdmaMeas.ooSync.set_timeout(timeout = 1.0)
```

No command help available

param timeout
No help available

Cloning the Group

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

Subgroups

6.1.1.5.1 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 LimitCls

Limit commands group definition. 3 total commands, 0 Subgroups, 3 group commands

get_poff_upper() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OOSync:LIMit:POFFupper
value: float = driver.configure.wcdmaMeas.ooSync.limit.get_poff_upper()
```

No command help available

```
return
po_ulimit: No help available
```

get_pon_upper() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OOSync:LIMit:PONupper
value: float = driver.configure.wcdmaMeas.ooSync.limit.get_pon_upper()
```

No command help available

```
return
pon_lower: No help available
```

get_threshold() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OOSync:LIMit:THReshold
value: float = driver.configure.wcdmaMeas.ooSync.limit.get_threshold()
```

No command help available

```
return
threshold_level: No help available
```

set_poff_upper(po_ulimit: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OOSync:LIMit:POFFupper
driver.configure.wcdmaMeas.ooSync.limit.set_poff_upper(po_ulimit = 1.0)
```

No command help available

param po_ulimit

No help available

set_pon_upper(pon_lower: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OOSync:LIMit:PONupper
driver.configure.wcdmaMeas.ooSync.limit.set_pon_upper(pon_lower = 1.0)
```

No command help available

param pon_lower

No help available

set_threshold(threshold_level: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:OOSync:LIMit:THReshold
driver.configure.wcdmaMeas.ooSync.limit.set_threshold(threshold_level = 1.0)
```

No command help available

param threshold_level

No help available

6.1.1.6 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 PrachCls

Prach commands group definition. 30 total commands, 2 Subgroups, 5 group commands

get_mo_exception() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:MOEXception
value: bool = driver.configure.wcdmaMeas.prach.get_mo_exception()
```

Specifies whether measurement results that the CMP180 identifies as faulty or inaccurate are rejected.

return

meas_on_exception: OFF: Faulty results are rejected. ON: Results are never rejected.

get_mpreamble() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:MPReamble
value: int = driver.configure.wcdmaMeas.prach.get_mpreamble()
```

Specifies the number of preambles to be measured.

return

preambles: No help available

get_off_power() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:OFFPower
value: bool = driver.configure.wcdmaMeas.prach.get_off_power()
```

Enables or disables the measurement of the off power before and after the last preamble.

return
enable: No help available

get_ppreamble() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:PPreamble
value: int = driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas. Configure.WcdmaMeas.Prach.mpreamble) .

return
preamble: No help available

get_timeout() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:TOUT
value: float = driver.configure.wcdmaMeas.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. 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: No help available

set_mo_exception(meas_on_exception: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:MOException
driver.configure.wcdmaMeas.prach.set_mo_exception(meas_on_exception = False)
```

Specifies whether measurement results that the CMP180 identifies as faulty or inaccurate are rejected.

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

set_mpreamble(preambles: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:MPreamble
driver.configure.wcdmaMeas.prach.set_mpreamble(preambles = 1)
```

Specifies the number of preambles to be measured.

param preambles
No help available

set_off_power(*enable: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:OFFPower
driver.configure.wcdmaMeas.prach.set_off_power(enable = False)
```

Enables or disables the measurement of the off power before and after the last preamble.

param enable

No help available

set_ppreamble(*preamble: int*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:PPreamble
driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas. Configure.WcdmaMeas.Prach.mpreamble) .

param preamble

No help available

set_timeout(*timeout: float*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:TOUT
driver.configure.wcdmaMeas.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. 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

No help available

Cloning the Group

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

Subgroups

6.1.1.6.1 Limit

SCPI Commands :

```
CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:IQOffset
CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:IQIMbalance
CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:CFERror
```

class LimitCls

Limit commands group definition. 13 total commands, 4 Subgroups, 3 group commands

get_cf_error() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:CFERror
value: float or bool = driver.configure.wcdmaMeas.prach.limit.get_cf_error()
```

Defines an upper limit for the carrier frequency error.

```
return
    frequency_error: (float or boolean) No help available
```

get_iq_imbalance() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:IQIMbalance
value: float or bool = driver.configure.wcdmaMeas.prach.limit.get_iq_imbalance()
```

Defines an upper limit for the I/Q imbalance.

```
return
    iq_imbalance: (float or boolean) No help available
```

get_iq_offset() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:IQOFfset
value: float or bool = driver.configure.wcdmaMeas.prach.limit.get_iq_offset()
```

Defines an upper limit for the I/Q origin offset.

```
return
    iq_offset: (float or boolean) No help available
```

set_cf_error(frequency_error: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:CFERror
driver.configure.wcdmaMeas.prach.limit.set_cf_error(frequency_error = 1.0)
```

Defines an upper limit for the carrier frequency error.

```
param frequency_error
    (float or boolean) No help available
```

set_iq_imbalance(iq_imbalance: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:IQIMbalance
driver.configure.wcdmaMeas.prach.limit.set_iq_imbalance(iq_imbalance = 1.0)
```

Defines an upper limit for the I/Q imbalance.

```
param iq_imbalance
    (float or boolean) No help available
```

set_iq_offset(iq_offset: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:IQOFfset
driver.configure.wcdmaMeas.prach.limit.set_iq_offset(iq_offset = 1.0)
```

Defines an upper limit for the I/Q origin offset.

param iq_offset
(float or boolean) No help available

Cloning the Group

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

Subgroups

6.1.1.6.1.1 EvMagnitude

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:EVMagnitude
```

class EvMagnitudeCls

EvMagnitude commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class EvMagnitudeStruct

Response structure. Fields:

- Rms: float or bool: No parameter help available
- Peak: float or bool: No parameter help available

get() → EvMagnitudeStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:EVMagnitude
value: EvMagnitudeStruct = driver.configure.wcdmaMeas.prach.limit.evMagnitude.
    ↪ get()
```

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.

set(rms: float, peak: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:EVMagnitude
driver.configure.wcdmaMeas.prach.limit.evMagnitude.set(rms = 1.0, peak = 1.0)
```

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

param rms

(float or boolean) No help available

param peak

(float or boolean) No help available

6.1.1.6.1.2 Merror

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:PRCh:LIMit:MERRor
```

class MerrorCls

Merror commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class MerrorStruct

Response structure. Fields:

- Rms: float or bool: No parameter help available
- Peak: float or bool: No parameter help available

get() → MerrorStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRCh:LIMit:MERRor
value: MerrorStruct = driver.configure.wcdmaMeas.prach.limit.merror.get()
```

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.

set(rms: float, peak: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRCh:LIMit:MERRor
driver.configure.wcdmaMeas.prach.limit.merror.set(rms = 1.0, peak = 1.0)
```

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

param rms

(float or boolean) No help available

param peak

(float or boolean) No help available

6.1.1.6.1.3 Pcontrol

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:PRCh:LIMit:PCONtrol:OFFPower
```

class PcontrolCls

Pcontrol commands group definition. 7 total commands, 3 Subgroups, 1 group commands

get_off_power() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRCh:LIMit:PCONtrol:OFFPower
value: float or bool = driver.configure.wcdmaMeas.prach.limit.pcontrol.get_off_
↪power()
```

Defines an upper OFF power limit. It also enables or disables the limit check.

return

limit: (float or boolean) No help available

set_off_power(limit: float) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:PRACH:LIMit:PCONtrol:OFFPower
driver.configure.wcdmaMeas.prach.limit.pcontrol.set_off_power(limit = 1.0)
```

Defines an upper OFF power limit. It also enables or disables the limit check.

param limit

(float or boolean) No help available

Cloning the Group

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

Subgroups

6.1.1.6.1.4 MaxPower

SCPI Commands :

```
CONFIGure:WCDma:MEASurement<instance>:PRACH:LIMit:PCONtrol:MAXPower:URPClass
CONFIGure:WCDma:MEASurement<instance>:PRACH:LIMit:PCONtrol:MAXPower
CONFIGure:WCDma:MEASurement<instance>:PRACH:LIMit:PCONtrol:MAXPower:ACTive
```

class MaxPowerCls

MaxPower commands group definition. 4 total commands, 1 Subgroups, 3 group commands

class ActiveStruct

Structure for reading output parameters. Fields:

- Nominal_Max_Power: float: Nominal maximum output power of the UE
- Upper_Limit: float: Tolerance value for too high maximum UE power
- Lower_Limit: float: Tolerance value for too low maximum UE power

class MaxPowerStruct

Response structure. Fields:

- Enable: bool: Disables | enables the limit check.
- Active_Limit: enums.ActiveLimit: To use the limits defined by 3GPP, select the power class of the UE (PC1 to PC4 = power class 1, 2, 3, 3bis, 4) . For user-defined limit values, select USER and define the limits via [CMDLINKRESOLVED Configure.WcdmaMeas.Prach.Limit.Pcontrol.MaxPower.UserDefined#set CMDLINKRESOLVED].

get() → MaxPowerStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:PRACH:LIMit:PCONtrol:MAXPower
value: MaxPowerStruct = driver.configure.wcdmaMeas.prach.limit.pcontrol.
↳maxPower.get()
```

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 MaxPowerStruct structure arguments.

get_active() → ActiveStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↳:PRACH:LIMit:PCONtrol:MAXPower:Active
value: ActiveStruct = driver.configure.wcdmaMeas.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.wcdmaMeas.prach.limit.pcontrol.maxPower.get_urp_
↳class()
```

No command help available

return

enable: No help available

set(enable: bool, active_limit: ActiveLimit) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:PRACH:LIMit:PCONtrol:MAXPower
driver.configure.wcdmaMeas.prach.limit.pcontrol.maxPower.set(enable = False,
↳active_limit = enums.ActiveLimit.PC1)
```

Enables or disables the check of the maximum output power limits and selects the set of limit settings to be used.

param enable

Disables | enables the limit check.

param active_limit

To use the limits defined by 3GPP, select the power class of the UE (PC1 to PC4 = power class 1, 2, 3, 3bis, 4) . For user-defined limit values, select USER and define the limits via method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.Limit.Pcontrol.MaxPower.UserDefined.set.

set_urp_class(enable: bool) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>
↳:PRACH:LIMit:PCONtrol:MAXPower:URPClass
driver.configure.wcdmaMeas.prach.limit.pcontrol.maxPower.set_urp_class(enable =
↳False)
```

No command help available

param enable
No help available

Cloning the Group

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

Subgroups

6.1.1.6.1.5 UserDefined

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:MAXPower:UDEFined
```

class UserDefinedCls

UserDefined commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class UserDefinedStruct

Response structure. Fields:

- Nominal_Max_Power: float: Nominal maximum output power of the UE
- Upper_Limit: float: Tolerance value for too high maximum UE power
- Lower_Limit: float: Tolerance value for too low maximum UE power

get() → UserDefinedStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:PRACH:LIMit:PCONtrol:MAXPower:UDEFined
value: UserDefinedStruct = driver.configure.wcdmaMeas.prach.limit.pcontrol.
↳maxPower.userDefined.get()
```

Sets the user-defined maximum output power limits. To activate the usage of this limit set, see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.Prach.Limit.Pcontrol.MaxPower.set.

return

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

set(nominal_max_power: float, upper_limit: float, lower_limit: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:PRACH:LIMit:PCONtrol:MAXPower:UDEFined
driver.configure.wcdmaMeas.prach.limit.pcontrol.maxPower.userDefined.
↳set(nominal_max_power = 1.0, upper_limit = 1.0, lower_limit = 1.0)
```

Sets the user-defined maximum output power limits. To activate the usage of this limit set, see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.Prach.Limit.Pcontrol.MaxPower.set.

param nominal_max_power

Nominal maximum output power of the UE

param upper_limit

Tolerance value for too high maximum UE power

param lower_limit

Tolerance value for too low maximum UE power

6.1.1.6.1.6 OIPower**SCPI Command :**

```
CONFigure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:OLPower
```

class OlPowerCls

OlPower commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class OlPowerStruct

Response structure. Fields:

- Enable: bool: Disables | enables the limit check.
- Init_Preamble_Pwr: float: Initial preamble power
- Olp_Limit: float: Open loop power tolerance value.

get() → OlPowerStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:OLPower
value: OlPowerStruct = driver.configure.wcdmaMeas.prach.limit.pcontrol.olPower.
↳get()
```

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.

set(enable: bool, init_preamble_pwr: float, olp_limit: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:OLPower
driver.configure.wcdmaMeas.prach.limit.pcontrol.olPower.set(enable = False,
↳init_preamble_pwr = 1.0, olp_limit = 1.0)
```

Enables or disables the check of the open loop power limits and specifies these limits.

param enable

Disables | enables the limit check.

param init_preamble_pwr

Initial preamble power

param olp_limit

Open loop power tolerance value.

6.1.1.6.1.7 Pstep

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:PSTep
```

class PstepCls

Pstep commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PstepStruct

Response structure. Fields:

- Enable: bool: Disables | enables the limit check.
- Preamble_Pwr_Step: float: Expected preamble power step size.
- Pwr_Step_Limit: float: Preamble power step tolerance value.

get() → PstepStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:PSTep
value: PstepStruct = driver.configure.wcdmaMeas.prach.limit.pcontrol.pstep.get()
```

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(enable: bool, preamble_pwr_step: float, pwr_step_limit: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:PRACH:LIMit:PCONtrol:PSTep
driver.configure.wcdmaMeas.prach.limit.pcontrol.pstep.set(enable = False,
↳ preamble_pwr_step = 1.0, pwr_step_limit = 1.0)
```

Enables or disables the check of the preamble power step limits and specifies these limits.

param enable

Disables | enables the limit check.

param preamble_pwr_step

Expected preamble power step size.

param pwr_step_limit

Preamble power step tolerance value.

6.1.1.6.1.8 Perror

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:PRACH:LIMit:PERRor
```

class PerrorCls

Perror commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PerrorStruct

Response structure. Fields:

- Rms: float or bool: No parameter help available
- Peak: float or bool: No parameter help available

get() → PerrorStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:PERRor
value: PerrorStruct = driver.configure.wcdmaMeas.prach.limit.perror.get()
```

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 value.

return

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

set(rms: float, peak: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:LIMit:PERRor
driver.configure.wcdmaMeas.prach.limit.perror.set(rms = 1.0, peak = 1.0)
```

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 value.

param rms

(float or boolean) No help available

param peak

(float or boolean) No help available

6.1.1.6.2 Result**SCPI Commands :**

```
CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult:UEPower
CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult:PSTeps
CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult:FERRor
CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult[:ALL]
CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult:PERRor
CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult:EVMagnitude
CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult:MERRor
CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult:IQ
```

class ResultCls

Result commands group definition. 12 total commands, 1 Subgroups, 8 group commands

class AllStruct

Structure for setting input parameters. Fields:

- Enable_Ue_Power: bool: UE power OFF: Do not evaluate results. ON: Evaluate the results.
- Enable_Pow_Steps: bool: Power steps
- Enable_Freq_Error: bool: Frequency error
- Enable_Evm: bool: Error vector magnitude

- Enable_Mag_Error: bool: Magnitude error
- Enable_Phase_Err: bool: Phase error
- Enable_Ue_Pchip: bool: UE power vs chip
- Enable_Evm_Chip: bool: EVM vs chip
- Enable_Merr_Chip: bool: Magnitude error vs chip
- Enable_Ph_Err_Chip: bool: Phase error vs chip
- Enable_Iq: bool: I/Q constellation diagram

get_all() → AllStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult[:ALL]
value: AllStruct = driver.configure.wcdmaMeas.prach.result.get_all()
```

Enables or disables the evaluation of results of the PRACH measurement. This command combines all other CONFIGure:WCDMa:MEAS<i><instance></i>:PRCh:RESult... commands.

return

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

get_ev_magnitude() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult:EVMagnitude
value: bool = driver.configure.wcdmaMeas.prach.result.get_ev_magnitude()
```

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

return

enable_evm: No help available

get_freq_error() → bool

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

return

enable_freq_error: No help available

get_iq() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRCh:RESult:IQ
value: bool = driver.configure.wcdmaMeas.prach.result.get_iq()
```

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

return
enable_iq: No help available

get_merror() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:MERRor
value: bool = driver.configure.wcdmaMeas.prach.result.get_merror()
```

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

return
enable_mag_error: No help available

get_perror() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:PERRor
value: bool = driver.configure.wcdmaMeas.prach.result.get_perror()
```

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

return
enable_phase_err: No help available

get_psteps() → bool

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

return
enable_pow_steps: No help available

get_ue_power() → bool

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

return
enable_ue_power: No help available

set_all(value: AllStruct) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult[:ALL]
structure = driver.configure.wcdmaMeas.prach.result.AllStruct()
structure.Enable_Ue_Power: bool = False
structure.Enable_Pow_Steps: bool = False
structure.Enable_Freq_Error: bool = False
structure.Enable_Evm: bool = False
structure.Enable_Mag_Error: bool = False
structure.Enable_Phase_Err: bool = False
structure.Enable_Ue_Pchip: bool = False
structure.Enable_Evm_Chip: bool = False
structure.Enable_Merr_Chip: bool = False
structure.Enable_Ph_Err_Chip: bool = False
structure.Enable_Iq: bool = False
driver.configure.wcdmaMeas.prach.result.set_all(value = structure)
```

Enables or disables the evaluation of results of the PRACH measurement. This command combines all other CONFIGure:WCDMa:MEAS<i><instance>: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.wcdmaMeas.prach.result.set_ev_magnitude(enable_evm = False)
```

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_evm

OFF: Do not evaluate results. ON: Evaluate the results.

set_freq_error(enable_freq_error: bool) → None

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_freq_error

OFF: Do not evaluate results. ON: Evaluate the results.

set_iq(enable_iq: bool) → None

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_iq

OFF: Do not evaluate results. ON: Evaluate the results.

set_merror(*enable_mag_error: bool*) → None

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_mag_error

OFF: Do not evaluate results. ON: Evaluate the results.

set_perror(*enable_phase_err: bool*) → None

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_phase_err

OFF: Do not evaluate results. ON: Evaluate the results.

set_psteps(*enable_pow_steps: bool*) → None

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_pow_steps

OFF: Do not evaluate results. ON: Evaluate the results.

set_ue_power(*enable_ue_power: bool*) → None

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_ue_power

OFF: Do not evaluate results. ON: Evaluate the results.

Cloning the Group

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

Subgroups

6.1.1.6.2.1 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 ChipCls

Chip commands group definition. 4 total commands, 0 Subgroups, 4 group commands

get_evm() → bool

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

```
return
    enable_evm_chip: No help available
```

get_merror() → bool

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

```
return
    enable_merr_chip: No help available
```

get_perror() → bool

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

return
 enable_ph_err_chip: No help available

get_ue_power() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:UEPower
value: bool = driver.configure.wcdmaMeas.prach.result.chip.get_ue_power()
```

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

return
 enable_ue_pchip: No help available

set_evm(enable_evm_chip: bool) → None

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_evm_chip
 OFF: Do not evaluate results. ON: Evaluate the results.

set_merror(enable_merr_chip: bool) → None

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

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_merr_chip
 OFF: Do not evaluate results. ON: Evaluate the results.

set_perror(enable_ph_err_chip: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:PRACH:RESult:CHIP:PERRor
driver.configure.wcdmaMeas.prach.result.chip.set_perror(enable_ph_err_chip =
False)
```

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_ph_err_chip
 OFF: Do not evaluate results. ON: Evaluate the results.

set_ue_power(enable_ue_pchip: bool) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:PRACH:RESult:CHIP:UEPower
driver.configure.wcdmaMeas.prach.result.chip.set_ue_power(enable_ue_pchip = False)
```

Enables or disables the evaluation of results of the PRACH measurement identified by the last command mnemonics: UE power, power steps, frequency error, error vector magnitude (EVM) , magnitude error, phase error, UE power vs chip, EVM vs chip, magnitude error vs chip, phase error vs chip, and I/Q constellation measurements.

param enable_ue_pchip

OFF: Do not evaluate results. ON: Evaluate the results.

6.1.1.7 RfSettings

SCPI Commands :

```
CONFIGure:WCDma:MEASurement<instance>:RFSettings:EATTenuation
CONFIGure:WCDma:MEASurement<instance>:RFSettings:UMARgin
CONFIGure:WCDma:MEASurement<instance>:RFSettings:ENPower
```

class RfSettingsCls

RfSettings commands group definition. 5 total commands, 2 Subgroups, 3 group commands

get_eattenuation() → float

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

Defines an external attenuation (or gain, if the value is negative) , to be applied to the input connector.

return

rf_input_ext_att: No help available

get_envelope_power() → float

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

Sets the expected nominal power of the measured RF signal.

return

exp_nom_power: 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 specifications document.

get_umargin() → float

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:RFSettings:UMARgin
value: float = driver.configure.wcdmaMeas.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 specifications document.

return

user_margin: No help available

set_eattenuation(rf_input_ext_att: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:RFSettings:EATTenuation
driver.configure.wcdmaMeas.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.

param rf_input_ext_att

No help available

set_envelope_power(exp_nom_power: float) → None

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

Sets the expected nominal power of the measured RF signal.

param exp_nom_power

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 specifications document.

set_umargin(user_margin: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:RFSettings:UMArgin
driver.configure.wcdmaMeas.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 specifications document.

param user_margin

No help available

Cloning the Group

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

Subgroups

6.1.1.7.1 Carrier<Carrier>

RepCap Settings

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

class CarrierCls

Carrier commands group definition. 1 total commands, 1 Subgroups, 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.wcdmaMeas.rfSettings.carrier.clone()
```

Subgroups**6.1.1.7.1.1 Frequency****SCPI Command :**

```
CONFigure:WCDma:MEASurement<instance>:RFSettings:CARRier<carrier>:FREQuency
```

class FrequencyCls

Frequency commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(carrier=Carrier.Default) → float

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

Selects the center frequency of the RF analyzer. For the supported frequency range, see ‘Frequency ranges’.

param carrier

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

return

frequency: No help available

set(frequency: float, carrier=Carrier.Default) → None

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

Selects the center frequency of the RF analyzer. For the supported frequency range, see ‘Frequency ranges’.

param frequency

No help available

param carrier

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

6.1.1.7.2 Dcarrier

SCPI Command :

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

class DcarrierCls

Dcarrier commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_separation() → float

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

Sets the carrier 1 and carrier 2 frequency separation for measurements with dual uplink carrier.

return
dc_freq_sep: No help available

set_separation(dc_freq_sep: float) → None

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

Sets the carrier 1 and carrier 2 frequency separation for measurements with dual uplink carrier.

param dc_freq_sep
No help available

6.1.1.8 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 TpcCls

Tpc commands group definition. 32 total commands, 7 Subgroups, 5 group commands

get_cselection() → Carrier

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:CSElection
value: enums.Carrier = driver.configure.wcdmaMeas.tpc.get_cselection()
```

Selects the uplink carrier to be measured. This parameter is relevant only for the dual uplink carrier configuration.

return
carrier: C1: primary uplink carrier C2: secondary uplink carrier

get_mo_exception() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:MOEXception
value: bool = driver.configure.wcdmaMeas.tpc.get_mo_exception()
```

Specifies whether measurement results that the CMP180 identifies as faulty or inaccurate are rejected.

return

meas_on_exception: OFF: Faulty results are rejected. ON: Results are never rejected.

get_mode() → MeasMode

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:MODE
value: enums.MeasMode = driver.configure.wcdmaMeas.tpc.get_mode()
```

Queries the active measurement mode resulting from the currently selected TPC setup.

return

meas_mode: 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() → SetType

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:SETup
value: enums.SetType = driver.configure.wcdmaMeas.tpc.get_setup()
```

Selects the TPC setup (expected) to be executed during the measurement.

return

set_type: 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: CONFIGure:WCDMa:MEASurement<instance>:TPC:TOUT
value: float = driver.configure.wcdmaMeas.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. 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: No help available

set_cselection(carrier: Carrier) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:CSElection
driver.configure.wcdmaMeas.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: primary uplink carrier C2: secondary uplink carrier

set_mo_exception(*meas_on_exception: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:MOEXception
driver.configure.wcdmaMeas.tpc.set_mo_exception(meas_on_exception = False)
```

Specifies whether measurement results that the CMP180 identifies as faulty or inaccurate are rejected.

param meas_on_exception

OFF: Faulty results are rejected. ON: Results are never rejected.

set_setup(*set_type: SetType*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:SETup
driver.configure.wcdmaMeas.tpc.set_setup(set_type = enums.SetType.ALL0)
```

Selects the TPC setup (expected) to be executed during the measurement.

param set_type

CLOop: Closed Loop ALTErning: 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.wcdmaMeas.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. 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

No help available

Cloning the Group

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

Subgroups

6.1.1.8.1 Ctfc

class CtfcCls

Ctfc commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1.8.1.1 Mlength

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:TPC:CTFC:MLENgtH
```

class MlengthCls

Mlength commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class GetStruct

Response structure. Fields:

- Nr_Steps: int: Number of steps to be measured per direction.
- Meas_Length: int: Number of slots to be measured.

get() → GetStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:CTFC:MLENgtH
value: GetStruct = driver.configure.wcdmaMeas.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.wcdmaMeas.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

Number of steps to be measured per direction.

6.1.1.8.2 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 DhibCls

Dhib commands group definition. 3 total commands, 0 Subgroups, 3 group commands

get_aexecution() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:AEExecution
value: bool = driver.configure.wcdmaMeas.tpc.dhib.get_aexecution()
```

No command help available

return

enable: No help available

get_mlength() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:MLENgtH
value: int = driver.configure.wcdmaMeas.tpc.dhib.get_mlength()
```

Defines the number of slots to be measured in DC HSDPA In-Band Emission mode.

return

meas_length: No help available

get_pattern() → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:PATtern
value: float = driver.configure.wcdmaMeas.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: UD: C1: 11... C2: 00... DU: C1: 00... C2: 11...

set_aexecution(enable: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:AEExecution
driver.configure.wcdmaMeas.tpc.dhib.set_aexecution(enable = False)
```

No command help available

param enable

No help available

set_mlength(*meas_length: int*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:MLENgtH
driver.configure.wcdmaMeas.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

No help available

set_pattern(*pattern: float*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:DHIB:PATtern
driver.configure.wcdmaMeas.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

UD: C1: 11... C2: 00... DU: C1: 00... C2: 11...

6.1.1.8.3 IlpControl

SCPI Commands :

```
CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:MLENgtH
CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSsegment
CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:AEExecution
```

class IlpControlCls

IlpControl commands group definition. 5 total commands, 2 Subgroups, 3 group commands

get_aexecution() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:AEExecution
value: bool = driver.configure.wcdmaMeas.tpc.ilpControl.get_aexecution()
```

No command help available

return

enable: No help available

get_mlength() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:MLENgtH
value: int = driver.configure.wcdmaMeas.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: No help available

get_ts_segment() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSSegment
value: bool = driver.configure.wcdmaMeas.tpc.ilpControl.get_ts_segment()
```

Enables or disables segmentation for test steps E, F, G and H.

return

enable: No help available

set_aexecution(enable: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:AEXecution
driver.configure.wcdmaMeas.tpc.ilpControl.set_aexecution(enable = False)
```

No command help available

param enable

No help available

set_ts_segment(enable: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSSegment
driver.configure.wcdmaMeas.tpc.ilpControl.set_ts_segment(enable = False)
```

Enables or disables segmentation for test steps E, F, G and H.

param enable

No help available

Cloning the Group

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

Subgroups

6.1.1.8.3.1 Tsef

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSEF
```

class TsefCls

Tsef commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class TsefStruct

Response structure. Fields:

- Length: int: Number of TPC bits per test step
- Statistics: int: Number of slots at the end of test step E (F) , where the minimum (maximum) output power results are measured.

get() → TsefStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSEF
value: TsefStruct = driver.configure.wcdmaMeas.tpc.ilpControl.tsef.get()
```

Configures the inner loop power control test steps E and F.

return

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

set(length: int, statistics: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSEF
driver.configure.wcdmaMeas.tpc.ilpControl.tsef.set(length = 1, statistics = 1)
```

Configures the inner loop power control test steps E and F.

param length

Number of TPC bits per test step

param statistics

Number of slots at the end of test step E (F) , where the minimum (maximum) output power results are measured.

6.1.1.8.3.2 Tsgh

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSGH
```

class TsghCls

Tsgh commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class TsghStruct

Response structure. Fields:

- Length: int: Number of TPC bits per test step
- Statistics: int: Number of slots at the end of test step G (H) , where the minimum (maximum) output power results are measured.

get() → TsghStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSGH
value: TsghStruct = driver.configure.wcdmaMeas.tpc.ilpControl.tsgh.get()
```

Configures the inner loop power control test steps G and H. For Signal Path = Network, use method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.Tpc.IlpControl.Tsgh.set.

return

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

set(length: int, statistics: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ILPControl:TSGH
driver.configure.wcdmaMeas.tpc.ilpControl.tsgh.set(length = 1, statistics = 1)
```

Configures the inner loop power control test steps G and H. For Signal Path = Network, use method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.Tpc.IlpControl.Tsgh.set.

param length

Number of TPC bits per test step

param statistics

Number of slots at the end of test step G (H) , where the minimum (maximum) output power results are measured.

6.1.1.8.4 Limit

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:DHIB
```

class LimitCls

Limit commands group definition. 13 total commands, 4 Subgroups, 1 group commands

get_dhib() → float

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:DHIB
value: float = driver.configure.wcdmaMeas.tpc.limit.get_dhib()
```

No command help available

return

min_power: No help available

set_dhib(min_power: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:DHIB
driver.configure.wcdmaMeas.tpc.limit.set_dhib(min_power = 1.0)
```

No command help available

param min_power

No help available

Cloning the Group

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

Subgroups

6.1.1.8.4.1 Ctfc

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:CTFC
```

class CtfcCls

Ctfc commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class CtfcStruct

Response structure. Fields:

- **Power_Step_Limit:** float: Symmetrical tolerance value for the power step size.
- **Calc_Beta_Factors:** bool: Enables or disables the automatic calculation of the expected power step size from the configured beta factors.
- **Power_Step_Size:** float: The expected power step size applicable if the automatic calculation from beta factors is disabled.

get() → CtfcStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:CTFC
value: CtfcStruct = driver.configure.wcdmaMeas.tpc.limit.ctfc.get()
```

Configures a power step limit for the measurement mode Change of TFC.

return

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

set(*power_step_limit: float, calc_beta_factors: bool, power_step_size: float = None*) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:CTFC
driver.configure.wcdmaMeas.tpc.limit.ctfc.set(power_step_limit = 1.0, calc_beta_
  factors = False, power_step_size = 1.0)
```

Configures a power step limit for the measurement mode Change of TFC.

param power_step_limit

Symmetrical tolerance value for the power step size.

param calc_beta_factors

Enables or disables the automatic calculation of the expected power step size from the configured beta factors.

param power_step_size

The expected power step size applicable if the automatic calculation from beta factors is disabled.

6.1.1.8.4.2 IlpControl

class IlpControlCls

IlpControl commands group definition. 8 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1.8.4.3 EpStep

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:EPStep
```

class EpStepCls

EpStep commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class EpStepStruct

Response structure. Fields:

- Enable: bool: No parameter help available
- Max_Count: int: Maximum allowed exceptions for sections BC, EF and GH
- Step_1_Db: float: Exceptional limit for step size 1 dB
- Step_2_Db: float: Exceptional limit for step size 2 dB

get() → EpStepStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:EPStep
value: EpStepStruct = driver.configure.wcdmaMeas.tpc.limit.ilpControl.epStep.
    ↪ get()
```

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.

set(enable: bool, max_count: int, step_1_db: float, step_2_db: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:EPStep
driver.configure.wcdmaMeas.tpc.limit.ilpControl.epStep.set(enable = False, max_
    ↪ count = 1, step_1_db = 1.0, step_2_db = 1.0)
```

Defines Inner Loop Power Control limits for exceptions and enables or disables the limit check.

param enable

No help available

param max_count

Maximum allowed exceptions for sections BC, EF and GH

param step_1_db

Exceptional limit for step size 1 dB

param step_2_db

Exceptional limit for step size 2 dB

6.1.1.8.4.4 MaxPower**SCPI Commands :**

```

CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MAXPower:URPClass
CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MAXPower
CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MAXPower:ACTIVE

```

class MaxPowerCls

MaxPower commands group definition. 4 total commands, 1 Subgroups, 3 group commands

class ActiveStruct

Structure for reading output parameters. Fields:

- Nominal_Max_Power: float: Nominal maximum output power of the UE
- Upper_Limit: float: Tolerance value for too high maximum UE power
- Lower_Limit: float: Tolerance value for too low maximum UE power

class MaxPowerStruct

Response structure. Fields:

- Enable: bool: Disables | enables the limit check.
- Active_Limit: enums.ActiveLimit: To use the limits defined by 3GPP, select the power class of the UE (PC1 to PC4 = power class 1, 2, 3, 3bis, 4) . For user-defined limit values, select USER and define the limits via [CMDLINKRESOLVED Configure.WcdmaMeas.Tpc.Limit.IlpControl.MaxPower.UserDefined#set CMDLINKRESOLVED].

get() → MaxPowerStruct

```

# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MAXPower
value: MaxPowerStruct = driver.configure.wcdmaMeas.tpc.limit.ilpControl.
↳maxPower.get()

```

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 MaxPowerStruct structure arguments.

get_active() → ActiveStruct

```

# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:TPC:LIMit:ILPControl:MAXPower:ACTIVE
value: ActiveStruct = driver.configure.wcdmaMeas.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.wcdmaMeas.tpc.limit.ilpControl.maxPower.get_urp_
↳class()
```

No command help available

return

enable: No help available

set(enable: bool, active_limit: ActiveLimit) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MAXPower
driver.configure.wcdmaMeas.tpc.limit.ilpControl.maxPower.set(enable = False,
↳active_limit = enums.ActiveLimit.PC1)
```

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 enable

Disables | enables the limit check.

param active_limit

To use the limits defined by 3GPP, select the power class of the UE (PC1 to PC4 = power class 1, 2, 3, 3bis, 4) . For user-defined limit values, select USER and define the limits via method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Tpc.Limit.IlpControl.MaxPower.UserDefined.set.

set_urp_class(enable: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:TPC:LIMit:ILPControl:MAXPower:URPClass
driver.configure.wcdmaMeas.tpc.limit.ilpControl.maxPower.set_urp_class(enable =
↳False)
```

No command help available

param enable

No help available

Cloning the Group

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

Subgroups

6.1.1.8.4.5 UserDefined

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MAXPower:UDEFined
```

class UserDefinedCls

UserDefined commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class UserDefinedStruct

Response structure. Fields:

- Nominal_Max_Power: float: Nominal maximum output power of the UE
- Upper_Limit: float: Tolerance value for too high maximum UE power
- Lower_Limit: float: Tolerance value for too low maximum UE power

get() → UserDefinedStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:TPC:LIMit:ILPControl:MAXPower:UDEFined
value: UserDefinedStruct = driver.configure.wcdmaMeas.tpc.limit.ilpControl.
↳maxPower.userDefined.get()
```

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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Tpc.Limit.IlpControl.MaxPower.set.

return

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

set(nominal_max_power: float, upper_limit: float, lower_limit: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>
↳:TPC:LIMit:ILPControl:MAXPower:UDEFined
driver.configure.wcdmaMeas.tpc.limit.ilpControl.maxPower.userDefined.
↳set(nominal_max_power = 1.0, upper_limit = 1.0, lower_limit = 1.0)
```

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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Tpc.Limit.IlpControl.MaxPower.set.

param nominal_max_power

Nominal maximum output power of the UE

param upper_limit

Tolerance value for too high maximum UE power

param lower_limit

Tolerance value for too low maximum UE power

6.1.1.8.4.6 MinPower**SCPI Command :**

```
CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MINPower
```

class MinPowerCls

MinPower commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class MinPowerStruct

Response structure. Fields:

- Enable: bool: Disables | enables the limit check.
- Upper_Limit: float: No parameter help available

get() → MinPowerStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MINPower
value: MinPowerStruct = driver.configure.wcdmaMeas.tpc.limit.ilpControl.
↳minPower.get()
```

Defines an Inner Loop Power Control limit: upper limit for the minimum UE output power. Also it enables or disables the limit check.

return

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

set(enable: bool, upper_limit: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:MINPower
driver.configure.wcdmaMeas.tpc.limit.ilpControl.minPower.set(enable = False,
↳upper_limit = 1.0)
```

Defines an Inner Loop Power Control limit: upper limit for the minimum UE output power. Also it enables or disables the limit check.

param enable

Disables | enables the limit check.

param upper_limit

No help available

6.1.1.8.4.7 PsGroup

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSGRoup
```

class PsGroupCls

PsGroup commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PsGroupStruct

Response structure. Fields:

- **Enable**: bool: Disables | enables the limit check.
- **Group_10_X_0_Db**: float: Limit for groups with expected step size 10 x 0 dB (algorithm 2) .
- **Group_10_X_1_Dba_Lg_2**: float: Limit for groups with expected step size 10 x ± 1 dB + 40 x 0 dB (algorithm 2) .
- **Group_10_X_1_Db**: float: Limit for groups with expected step size 10 x ± 1 dB (algorithm 1) .
- **Group_10_X_2_Db**: float: Limit for groups with expected step size 10 x ± 2 dB (algorithm 1) .

get() → PsGroupStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSGRoup
value: PsGroupStruct = driver.configure.wcdmaMeas.tpc.limit.ilpControl.psGroup.
↳get()
```

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 it enables or disables the limit check.

return

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

set(enable: bool, group_10_x_0_db: float, group_10_x_1_dba_lg_2: float, group_10_x_1_db: float, group_10_x_2_db: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSGRoup
driver.configure.wcdmaMeas.tpc.limit.ilpControl.psGroup.set(enable = False,
↳group_10_x_0_db = 1.0, group_10_x_1_dba_lg_2 = 1.0, group_10_x_1_db = 1.0,
↳group_10_x_2_db = 1.0)
```

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 it enables or disables the limit check.

param enable

Disables | enables the limit check.

param group_10_x_0_db

Limit for groups with expected step size 10 x 0 dB (algorithm 2) .

param group_10_x_1_dba_lg_2

Limit for groups with expected step size 10 x ± 1 dB + 40 x 0 dB (algorithm 2) .

param group_10_x_1_db

Limit for groups with expected step size 10 x ± 1 dB (algorithm 1) .

param group_10_x_2_db

Limit for groups with expected step size 10 x ± 2 dB (algorithm 1) .

6.1.1.8.4.8 Pstep

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSTep
```

class PstepCls

Pstep commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PstepStruct

Response structure. Fields:

- Enable: bool: Disables | enables the limit check.
- Step_0_Db: float: Limit for steps with expected step size 0 dB.
- Step_1_Db: float: Limit for steps with expected step size ± 1 dB.
- Step_2_Db: float: Limit for steps with expected step size ± 2 dB.

get() → PstepStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSTep
value: PstepStruct = driver.configure.wcdmaMeas.tpc.limit.ilpControl.pstep.get()
```

Defines Inner Loop Power Control limits: upper limits for the absolute value of the power step error, depending on the expected step size. Also it enables or disables the limit check.

return

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

set(enable: bool, step_0_db: float, step_1_db: float, step_2_db: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:ILPControl:PSTep
driver.configure.wcdmaMeas.tpc.limit.ilpControl.pstep.set(enable = False, step_
0_db = 1.0, step_1_db = 1.0, step_2_db = 1.0)
```

Defines Inner Loop Power Control limits: upper limits for the absolute value of the power step error, depending on the expected step size. Also it enables or disables the limit check.

param enable

Disables | enables the limit check.

param step_0_db

Limit for steps with expected step size 0 dB.

param step_1_db

Limit for steps with expected step size ± 1 dB.

param step_2_db

Limit for steps with expected step size ± 2 dB.

6.1.1.8.4.9 Mpedch

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:MPEDch
```

class MpedchCls

Mpedch commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class MpedchStruct

Response structure. Fields:

- **Enable**: bool: Disables | enables the limit check.
- **Nom_Max_Power**: float: Nominal maximum UE power.
- **Upper_Limit**: float: Upper limit = nominal power + this value.
- **Lower_Limit**: float: Lower limit = nominal power + this value.

get() → MpedchStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:MPEDch
value: MpedchStruct = driver.configure.wcdmaMeas.tpc.limit.mpedch.get()
```

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(enable: bool, nom_max_power: float, upper_limit: float, lower_limit: float) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:LIMit:MPEDch
driver.configure.wcdmaMeas.tpc.limit.mpedch.set(enable = False, nom_max_power = 1.0,
upper_limit = 1.0, lower_limit = 1.0)
```

Configures UE power limits for the measurement mode Max. Power E-DCH.

param enable

Disables | enables the limit check.

param nom_max_power

Nominal maximum UE power.

param upper_limit

Upper limit = nominal power + this value.

param lower_limit

Lower limit = nominal power + this value.

6.1.1.8.4.10 Ulcm

class UlcmCls

Ulcm commands group definition. 2 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.1.1.8.4.11 Pa

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PA
```

class PaCls

Pa commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PaStruct

Response structure. Fields:

- Initial_Pwr_Step: float or bool: Symmetrical tolerance value for UE TX power in the first slot after the gap
- Power_Step: float or bool: Symmetrical tolerance value for UE TX power in a recovery period
- Power_Step_Group: float or bool: Symmetrical tolerance value for the aggregate UE TX power in the recovery period comprising the 7 rising or falling power steps after each gap

get() → PaStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PA
value: PaStruct = driver.configure.wcdmaMeas.tpc.limit.ulcm.pa.get()
```

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.

set(initial_pwr_step: float, power_step: float, power_step_group: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PA
driver.configure.wcdmaMeas.tpc.limit.ulcm.pa.set(initial_pwr_step = 1.0, power_
↪step = 1.0, power_step_group = 1.0)
```

Configures a power step limit for the measurement mode UL Compressed Mode, CM pattern A.

param initial_pwr_step

(float or boolean) Symmetrical tolerance value for UE TX power in the first slot after the gap

param power_step

(float or boolean) Symmetrical tolerance value for UE TX power in a recovery period

param power_step_group

(float or boolean) Symmetrical tolerance value for the aggregate UE TX power in the recovery period comprising the 7 rising or falling power steps after each gap

6.1.1.8.4.12 Pb**SCPI Command :**

```
CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PB
```

class PbCls

Pb commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class PbStruct

Response structure. Fields:

- Initial_Pwr_Step: float or bool: Symmetrical tolerance value for the UE TX power in the first slot after the gap
- Power_Step: float or bool: Symmetrical tolerance value for the UE TX power in the nonCM - CM and CM - nonCM power step

get() → PbStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PB
value: PbStruct = driver.configure.wcdmaMeas.tpc.limit.ulcm.pb.get()
```

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(initial_pwr_step: float, power_step: float) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:LIMit:ULCM:PB
driver.configure.wcdmaMeas.tpc.limit.ulcm.pb.set(initial_pwr_step = 1.0, power_
↪step = 1.0)
```

Configures a power step limit for the measurement mode UL Compressed Mode, CM pattern B.

param initial_pwr_step

(float or boolean) Symmetrical tolerance value for the UE TX power in the first slot after the gap

param power_step

(float or boolean) Symmetrical tolerance value for the UE TX power in the nonCM - CM and CM - nonCM power step

6.1.1.8.5 Monitor

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:TPC:MONitor:MLENgtH
```

class MonitorCls

Monitor commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_mlength() → int

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MONitor:MLENgtH
value: int = driver.configure.wcdmaMeas.tpc.monitor.get_mlength()
```

Defines the number of slots to be measured in Monitor mode.

return
meas_length: No help available

set_mlength(meas_length: int) → None

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MONitor:MLENgtH
driver.configure.wcdmaMeas.tpc.monitor.set_mlength(meas_length = 1)
```

Defines the number of slots to be measured in Monitor mode.

param meas_length
No help available

6.1.1.8.6 Mpedch

SCPI Commands :

```
CONFigure:WCDMa:MEASurement<instance>:TPC:MPEDch:MLENgtH
CONFigure:WCDMa:MEASurement<instance>:TPC:MPEDch:AEXecution
```

class MpedchCls

Mpedch commands group definition. 2 total commands, 0 Subgroups, 2 group commands

get_aexecution() → bool

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MPEDch:AEXecution
value: bool = driver.configure.wcdmaMeas.tpc.mpedch.get_aexecution()
```

No command help available

return
enable: No help available

get_mlength() → int

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:TPC:MPEDch:MLENgtH
value: int = driver.configure.wcdmaMeas.tpc.mpedch.get_mlength()
```

Defines the number of slots to be measured in Max. Power E-DCH mode.

return
meas_length: No help available

set_aexecution(enable: bool) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:MPEDch:AEXecution
driver.configure.wcdmaMeas.tpc.mpedch.set_aexecution(enable = False)
```

No command help available

param enable
No help available

set_mlength(meas_length: int) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:MPEDch:MLENgtH
driver.configure.wcdmaMeas.tpc.mpedch.set_mlength(meas_length = 1)
```

Defines the number of slots to be measured in Max. Power E-DCH mode.

param meas_length
No help available

6.1.1.8.7 Ulcm

SCPI Commands :

```
CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:MLENgtH
CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:AEXecution
```

class UlcmCls

Ulcm commands group definition. 2 total commands, 0 Subgroups, 2 group commands

get_aexecution() → bool

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:AEXecution
value: bool = driver.configure.wcdmaMeas.tpc.ulcm.get_aexecution()
```

No command help available

return
enable: No help available

get_mlength() → int

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:MLENgtH
value: int = driver.configure.wcdmaMeas.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: No help available

set_aexecution(*enable: bool*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:AEExecution
driver.configure.wcdmaMeas.tpc.ulcm.set_aexecution(enable = False)
```

No command help available

param enable

No help available

set_mlength(*meas_length: int*) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:TPC:ULCM:MLENgtH
driver.configure.wcdmaMeas.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

No help available

6.1.1.9 UeChannels

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:UECHannels:BSFSelection
```

class UeChannelsCls

UeChannels commands group definition. 7 total commands, 1 Subgroups, 1 group commands

get_bsf_selection() → AutoManualMode

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UECHannels:BSFSelection
value: enums.AutoManualMode = driver.configure.wcdmaMeas.ueChannels.get_bsf_
↪selection()
```

No command help available

return

selection: No help available

set_bsf_selection(*selection: AutoManualMode*) → None

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

No command help available

param selection

No help available

Cloning the Group

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

Subgroups

6.1.1.9.1 Carrier<Carrier>

RepCap Settings

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

class CarrierCls

Carrier commands group definition. 6 total commands, 5 Subgroups, 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.wcdmaMeas.ueChannels.carrier.clone()
```

Subgroups

6.1.1.9.1.1 Dpcch

SCPI Command :

```
CONFigure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>:DPCCh
```

class DpcchCls

Dpcch commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class DpcchStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled
- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available

get(carrier=Carrier.Default) → DpcchStruct

```
# SCPI: CONFigure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>:DPCCh
value: DpcchStruct = driver.configure.wcdmaMeas.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.

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(enable: bool, beta_factor: int, spreading_factor: int, carrier=Carrier.Default) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>:DPCCh
driver.configure.wcdmaMeas.ueChannels.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.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

param carrier

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

6.1.1.9.1.2 Dpdch

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>:DPDCh
```

class DpdchCls

Dpdch commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class DpdchStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled
- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available

get(carrier=Carrier.Default) → DpdchStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>:DPDCh
value: DpdchStruct = driver.configure.wcdmaMeas.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.

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(*enable*: bool, *beta_factor*: int, *spreading_factor*: int, *carrier*=Carrier.Default) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:UECHannels:CARRier<carrier>:DPDCh
driver.configure.wcdmaMeas.ueChannels.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.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

param carrier

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

6.1.1.9.1.3 Edpcch

SCPI Command :

```
CONFIGure:WCDma:MEASurement<instance>:UECHannels:CARRier<carrier>:EDPCch
```

class EdpcchCls

Edpcch commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class EdpcchStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled
- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available

get(*carrier*=Carrier.Default) → EdpcchStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:UECHannels:CARRier<carrier>:EDPCch
value: EdpcchStruct = driver.configure.wcdmaMeas.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.

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(enable: bool, beta_factor: int, spreading_factor: int, carrier=Carrier.Default) → None

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:UEChannels:CARRier<carrier>:EDPCch
driver.configure.wcdmaMeas.ueChannels.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.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

param carrier

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

6.1.1.9.1.4 Edpdch<EdpdChannel>

RepCap Settings

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

SCPI Command :

```
CONFIGure:WCDma:MEASurement<instance>:UEChannels:CARRier<carrier>:EDPDch<nr>
```

class EdpdchCls

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

class EdpdchStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled
- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available

get(*carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → EdpdchStruct

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>:EDPDch
↳<nr>
value: EdpdchStruct = driver.configure.wcdmaMeas.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.

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(*enable*: bool, *beta_factor*: int, *spreading_factor*: int, *carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UECHannels:CARRier<carrier>:EDPDch
↳<nr>
driver.configure.wcdmaMeas.ueChannels.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.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

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.wcdmaMeas.ueChannels.carrier.edpdch.clone()
```

6.1.1.9.1.5 Hsdpcch

SCPI Command :

```
CONFIGure:WCDma:MEASurement<instance>:UECHannels:CARRier<carrier>:HSDPcch
```

class HsdpcchCls

Hsdpcch commands group definition. 2 total commands, 1 Subgroups, 1 group commands

class HsdpcchStruct

Response structure. Fields:

- Enable: bool: Channel disabled | enabled
- Beta_Factor: int: No parameter help available
- Spreading_Factor: int: No parameter help available

get(carrier=Carrier.Default) → HsdpcchStruct

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:UECHannels:CARRier<carrier>
↳:HSDPcch
value: HsdpcchStruct = driver.configure.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.UeChannels.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 HsdpcchStruct structure arguments.

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

```
# SCPI: CONFIGure:WCDma:MEASurement<instance>:UECHannels:CARRier<carrier>
↳:HSDPcch
driver.configure.wcdmaMeas.ueChannels.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. 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.UeChannels.Carrier.Hsdpcch.Config.set.

param enable

Channel disabled | enabled

param beta_factor

No help available

param spreading_factor

No help available

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.wcdmaMeas.ueChannels.carrier.hsdpcch.clone()
```

Subgroups**6.1.1.9.1.6 Config****SCPI Command :**

```
CONFIGure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>:HSDPcch:CONFig
```

class ConfigCls

Config commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(carrier=Carrier.Default) → Type

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>
↳:HSDPcch:CONFig
value: enums.Type = driver.configure.wcdmaMeas.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.

param carrier

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

return

type_py: No help available

set(type_py: Type, carrier=Carrier.Default) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UEChannels:CARRier<carrier>
↳:HSDPcch:CONFig
driver.configure.wcdmaMeas.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.

param type_py

No help available

param carrier

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

6.1.1.10 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 UeSignalCls

UeSignal commands group definition. 5 total commands, 1 Subgroups, 4 group commands

get_cm_pattern() → PatternType

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

Selects the expected TPC pattern for UL compressed mode.

return

pattern_type: 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.wcdmaMeas.ueSignal.get_dpdch()
```

Defines whether the UL DPCH contains a DPDCH.

return

dpdch: OFF: DPCCH only ON: DPCCH plus DPDCH

get_sformat() → int

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

Selects the slot format for the UL DPCCH.

return

slot_format: No help available

get_ul_config() → *UlConfiguration*

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

Selects the uplink signal configuration.

return

ul_configuration: 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

set_cm_pattern()(*pattern_type: PatternType*) → None

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

Selects the expected TPC pattern for UL compressed mode.

param pattern_type

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.wcdmaMeas.ueSignal.set_dpdch(dpdch = False)
```

Defines whether the UL DPCH contains a DPDCH.

param dpdch

OFF: DPCCH only ON: DPCCH plus DPDCH

set_sformat()(*slot_format: int*) → None

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

Selects the slot format for the UL DPCCH.

param slot_format

No help available

set_ul_config()(*ul_configuration: UlConfiguration*) → None

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

Selects the uplink signal configuration.

param ul_configuration

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

Cloning the Group

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

Subgroups

6.1.1.10.1 Carrier<Carrier>

RepCap Settings

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

class CarrierCls

Carrier commands group definition. 1 total commands, 1 Subgroups, 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.wcdmaMeas.ueSignal.carrier.clone()
```

Subgroups

6.1.1.10.1.1 Scode

SCPI Command :

```
CONFIGure:WCDMa:MEASurement<instance>:UESignal:CARRier<carrier>:SCODE
```

class ScodeCls

Scode commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get(carrier=Carrier.Default) → float

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:CARRier<carrier>:SCODE
value: float = driver.configure.wcdmaMeas.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.

param carrier

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

return

code: No help available

set(code: float, carrier=Carrier.Default) → None

```
# SCPI: CONFIGure:WCDMa:MEASurement<instance>:UESignal:CARRier<carrier>:SCODE
driver.configure.wcdmaMeas.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.

param code

No help available

param carrier

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

6.2 Route

class RouteCls

Route commands group definition. 5 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.2.1 WcdmaMeas

SCPI Command :

```
ROUTE:WCDMa:MEASurement<instance>
```

class WcdmaMeasCls

WcdmaMeas commands group definition. 5 total commands, 1 Subgroups, 1 group commands

class ValueStruct

Structure for reading output parameters. Fields:

- Scenario: enums.TestScenarioB: No parameter help available
- Controller: str: No parameter help available
- Rx_Connector: enums.RxConnector: No parameter help available
- Rx_Converter: enums.RxConverter: No parameter help available

get_value() → ValueStruct

```
# SCPI: ROUTe:WCDMa:MEASurement<instance>
value: ValueStruct = driver.route.wcdmaMeas.get_value()
```

No command help available

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.wcdmaMeas.clone()
```

Subgroups

6.2.1.1 Scenario

SCPI Commands :

```
ROUTE:WCDMa:MEASurement<instance>:SCENario:CSPath
ROUTE:WCDMa:MEASurement<instance>:SCENario
```

class ScenarioCls

Scenario commands group definition. 4 total commands, 2 Subgroups, 2 group commands

get_cspath() → str

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

No command help available

return

master: No help available

get_value() → TestScenarioB

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

No command help available

return

scenario: No help available

set_cspath(master: str) → None

```
# SCPI: ROUTe:WCDMa:MEASurement<instance>:SCENario:CSPath
driver.route.wcdmaMeas.scenario.set_cspath(master = 'abc')
```

No command help available

param master
No help available

Cloning the Group

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

Subgroups

6.2.1.1.1 MaProtocol

SCPI Command :

```
ROUTE:WCDMa:MEASurement<instance>:SCENario:MAProtocol
```

class MaProtocolCls

MaProtocol commands group definition. 1 total commands, 0 Subgroups, 1 group commands

set(*controler: str = None*) → None

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

No command help available

param controler
No help available

6.2.1.1.2 Salone

SCPI Command :

```
ROUTE:WCDMa:MEASurement<instance>:SCENario:SALone
```

class SaloneCls

Salone commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class SaloneStruct

Response structure. Fields:

- Rx_Connector: enums.RxConnector: No parameter help available
- Rf_Converter: enums.RxConverter: No parameter help available

get() → SaloneStruct

```
# SCPI: ROUTE:WCDMa:MEASurement<instance>:SCENario:SALone
value: SaloneStruct = driver.route.wcdmaMeas.scenario.salone.get()
```

No command help available

return

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

set(rx_connector: RxConnector, rf_converter: RxConverter) → None

```
# SCPI: ROUTe:WCDMa:MEASurement<instance>:SCENario:SALone
driver.route.wcdmaMeas.scenario.salone.set(rx_connector = enums.RxConnector.
↳ I11I, rf_converter = enums.RxConverter.IRX1)
```

No command help available

param rx_connector

No help available

param rf_converter

No help available

6.3 Trigger

class TriggerCls

Trigger commands group definition. 36 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.3.1 WcdmaMeas

class WcdmaMeasCls

WcdmaMeas commands group definition. 36 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.3.1.1 MultiEval

SCPI Commands :

```

TRIGger:WCDMa:MEASurement<instance>:MEvaluation:DElay
TRIGger:WCDMa:MEASurement<instance>:MEvaluation:MGAP
TRIGger:WCDMa:MEASurement<instance>:MEvaluation:THReshold
TRIGger:WCDMa:MEASurement<instance>:MEvaluation:SLOPe
TRIGger:WCDMa:MEASurement<instance>:MEvaluation:TOUT
TRIGger:WCDMa:MEASurement<instance>:MEvaluation:SOURce

```

class MultiEvalCls

MultiEval commands group definition. 8 total commands, 2 Subgroups, 6 group commands

get_delay() → float

```

# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEvaluation:DElay
value: float = driver.trigger.wcdmaMeas.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 always starts 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: No help available

get_mgap() → float

```

# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEvaluation:MGAP
value: float = driver.trigger.wcdmaMeas.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: No help available

get_slope() → SignalSlope

```

# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEvaluation:SLOPe
value: enums.SignalSlope = driver.trigger.wcdmaMeas.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: Rising edge FEDGe: Falling edge

get_source() → str

```

# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEvaluation:SOURce
value: str = driver.trigger.wcdmaMeas.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: '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.wcdmaMeas.multiEval.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return

level: No help available

get_timeout() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:TOUT
value: float or bool = driver.trigger.wcdmaMeas.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: (float or boolean) No help available

set_delay(delay: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:DElay
driver.trigger.wcdmaMeas.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 always starts 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

No help available

set_mgap(minimum_gap: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:MGAP
driver.trigger.wcdmaMeas.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

No help available

set_slope(slope: SignalSlope) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:SLOPe
driver.trigger.wcdmaMeas.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: Rising edge FEDGe: Falling edge

set_source(*source: str*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:SOURce
driver.trigger.wcdmaMeas.multiEval.set_source(source = 'abc')
```

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

‘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.wcdmaMeas.multiEval.set_threshold(level = 1.0)
```

Defines the trigger threshold for power trigger sources.

param level

No help available

set_timeout(*timeout: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:MEValuation:TOUT
driver.trigger.wcdmaMeas.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

(float or boolean) No help available

Cloning the Group

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

Subgroups

6.3.1.1.1 Catalog

SCPI Command :

```
TRIGger:WCDma:MEASurement<instance>:MEValuation:CATalog:SOURce
```

class CatalogCls

Catalog commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_source() → List[str]

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

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

return

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

6.3.1.1.2 ListPy

SCPI Command :

```
TRIGger:WCDma:MEASurement<instance>:MEValuation:LIST:MODE
```

class ListPyCls

ListPy commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_mode() → Mode

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:MEValuation:LIST:MODE
value: enums.Mode = driver.trigger.wcdmaMeas.multiEval.listPy.get_mode()
```

Specifies the trigger mode for list mode measurements. For configuration of retrigger flags, see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.Segment.Setup.set.

return

mode: 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: Mode) → None

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:MEValuation:LIST:MODE
driver.trigger.wcdmaMeas.multiEval.listPy.set_mode(mode = enums.Mode.ONCE)
```

Specifies the trigger mode for list mode measurements. For configuration of retrigger flags, see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.Segment.Setup.set.

param mode

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.

6.3.1.2 OlpControl**SCPI Commands :**

```
TRIGger:WCDMa:MEASurement<instance>:OLPControl:DElay
TRIGger:WCDMa:MEASurement<instance>:OLPControl:MGAP
TRIGger:WCDMa:MEASurement<instance>:OLPControl:THReshold
TRIGger:WCDMa:MEASurement<instance>:OLPControl:SLOPe
TRIGger:WCDMa:MEASurement<instance>:OLPControl:TOUT
TRIGger:WCDMa:MEASurement<instance>:OLPControl:SOURce
```

class OlpControlCls

OlpControl commands group definition. 7 total commands, 1 Subgroups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:DElay
value: float = driver.trigger.wcdmaMeas.olpControl.get_delay()
```

No command help available

return
delay: No help available

get_mgap() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:MGAP
value: float = driver.trigger.wcdmaMeas.olpControl.get_mgap()
```

No command help available

return
minimum_gap: No help available

get_slope() → SignalSlope

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:SLOPe
value: enums.SignalSlope = driver.trigger.wcdmaMeas.olpControl.get_slope()
```

No command help available

return
slope: No help available

get_source() → str

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:SOURce
value: str = driver.trigger.wcdmaMeas.olpControl.get_source()
```

No command help available

return

source: No help available

get_threshold() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:THReshold
value: float = driver.trigger.wcdmaMeas.olpControl.get_threshold()
```

No command help available

return

level: No help available

get_timeout() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:TOUT
value: float or bool = driver.trigger.wcdmaMeas.olpControl.get_timeout()
```

No command help available

return

timeout: (float or boolean) No help available

set_delay(delay: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:DElay
driver.trigger.wcdmaMeas.olpControl.set_delay(delay = 1.0)
```

No command help available

param delay

No help available

set_mgap(minimum_gap: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:MGAP
driver.trigger.wcdmaMeas.olpControl.set_mgap(minimum_gap = 1.0)
```

No command help available

param minimum_gap

No help available

set_slope(slope: SignalSlope) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:SLOPe
driver.trigger.wcdmaMeas.olpControl.set_slope(slope = enums.SignalSlope.FEDGe)
```

No command help available

param slope

No help available

set_source(source: str) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:SOURce
driver.trigger.wcdmaMeas.olpControl.set_source(source = 'abc')
```

No command help available

param source

No help available

set_threshold(*level: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:THReshold
driver.trigger.wcdmaMeas.olpControl.set_threshold(level = 1.0)
```

No command help available

param level

No help available

set_timeout(*timeout: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:TOUT
driver.trigger.wcdmaMeas.olpControl.set_timeout(timeout = 1.0)
```

No command help available

param timeout

(float or boolean) No help available

Cloning the Group

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

Subgroups

6.3.1.2.1 Catalog

SCPI Command :

```
TRIGger:WCDMa:MEASurement<instance>:OLPControl:CATalog:SOURce
```

class CatalogCls

Catalog commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_source() → List[str]

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OLPControl:CATalog:SOURce
value: List[str] = driver.trigger.wcdmaMeas.olpControl.catalog.get_source()
```

No command help available

return

trigger_list: No help available

6.3.1.3 OoSync

SCPI Commands :

```
TRIGger:WCDma:MEASurement<instance>:OOSync:DElay
TRIGger:WCDma:MEASurement<instance>:OOSync:MGAP
TRIGger:WCDma:MEASurement<instance>:OOSync:THReshold
TRIGger:WCDma:MEASurement<instance>:OOSync:SLOPe
TRIGger:WCDma:MEASurement<instance>:OOSync:TOUT
TRIGger:WCDma:MEASurement<instance>:OOSync:SOURce
```

class OoSyncCls

OoSync commands group definition. 7 total commands, 1 Subgroups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:OOSync:DElay
value: float = driver.trigger.wcdmaMeas.ooSync.get_delay()
```

No command help available

```
return
    delay: No help available
```

get_mgap() → float

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:OOSync:MGAP
value: float = driver.trigger.wcdmaMeas.ooSync.get_mgap()
```

No command help available

```
return
    minimum_gap: No help available
```

get_slope() → SignalSlope

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:OOSync:SLOPe
value: enums.SignalSlope = driver.trigger.wcdmaMeas.ooSync.get_slope()
```

No command help available

```
return
    slope: No help available
```

get_source() → str

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:OOSync:SOURce
value: str = driver.trigger.wcdmaMeas.ooSync.get_source()
```

No command help available

```
return
    source: No help available
```

get_threshold() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:THReshold
value: float = driver.trigger.wcdmaMeas.ooSync.get_threshold()
```

No command help available

return
level: No help available

get_timeout() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:TOUT
value: float or bool = driver.trigger.wcdmaMeas.ooSync.get_timeout()
```

No command help available

return
timeout: (float or boolean) No help available

set_delay(delay: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:DElay
driver.trigger.wcdmaMeas.ooSync.set_delay(delay = 1.0)
```

No command help available

param delay
No help available

set_mgap(minimum_gap: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:MGAP
driver.trigger.wcdmaMeas.ooSync.set_mgap(minimum_gap = 1.0)
```

No command help available

param minimum_gap
No help available

set_slope(slope: SignalSlope) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:SLOPe
driver.trigger.wcdmaMeas.ooSync.set_slope(slope = enums.SignalSlope.FEDGE)
```

No command help available

param slope
No help available

set_source(source: str) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:SOURce
driver.trigger.wcdmaMeas.ooSync.set_source(source = 'abc')
```

No command help available

param source
No help available

set_threshold(*level: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:THReshold
driver.trigger.wcdmaMeas.ooSync.set_threshold(level = 1.0)
```

No command help available

param level

No help available

set_timeout(*timeout: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:TOUT
driver.trigger.wcdmaMeas.ooSync.set_timeout(timeout = 1.0)
```

No command help available

param timeout

(float or boolean) No help available

Cloning the Group

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

Subgroups

6.3.1.3.1 Catalog

SCPI Command :

```
TRIGger:WCDMa:MEASurement<instance>:OOSync:CATalog:SOURce
```

class CatalogCls

Catalog commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_source() → List[str]

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:OOSync:CATalog:SOURce
value: List[str] = driver.trigger.wcdmaMeas.ooSync.catalog.get_source()
```

No command help available

return

trigger_list: No help available

6.3.1.4 Prach

SCPI Commands :

```
TRIGger:WCDMa:MEASurement<instance>:PRACH:DElay
TRIGger:WCDMa:MEASurement<instance>:PRACH:MGAP
TRIGger:WCDMa:MEASurement<instance>:PRACH:THReshold
TRIGger:WCDMa:MEASurement<instance>:PRACH:SLOPe
TRIGger:WCDMa:MEASurement<instance>:PRACH:TOUT
TRIGger:WCDMa:MEASurement<instance>:PRACH:SOURce
```

class PrachCls

Prach commands group definition. 7 total commands, 1 Subgroups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:DElay
value: float = driver.trigger.wcdmaMeas.prach.get_delay()
```

Defines a time delaying the start of the measurement relative to the trigger event.

return
delay: No help available

get_mgap() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:MGAP
value: float = driver.trigger.wcdmaMeas.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: No help available

get_slope() → SignalSlope

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:SLOPe
value: enums.SignalSlope = driver.trigger.wcdmaMeas.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: Rising edge FEDGe: Falling edge

get_source() → str

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:SOURce
value: str = driver.trigger.wcdmaMeas.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: 'IF Power (Sync) ': Power trigger (extended synchronization)

get_threshold() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:THReshold
value: float = driver.trigger.wcdmaMeas.prach.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return

level: No help available

get_timeout() → float

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:TOUT
value: float or bool = driver.trigger.wcdmaMeas.prach.get_timeout()
```

Selects the maximum time that the CMP180 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: (float or boolean) No help available

set_delay(delay: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:DElay
driver.trigger.wcdmaMeas.prach.set_delay(delay = 1.0)
```

Defines a time delaying the start of the measurement relative to the trigger event.

param delay

No help available

set_mgap(minimum_gap: float) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:MGAP
driver.trigger.wcdmaMeas.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

No help available

set_slope(slope: SignalSlope) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:SLOPe
driver.trigger.wcdmaMeas.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: Rising edge FEDGe: Falling edge

set_source(source: str) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:SOURce
driver.trigger.wcdmaMeas.prach.set_source(source = 'abc')
```

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

‘IF Power (Sync)’: Power trigger (extended synchronization)

set_threshold(*level: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:THReshold
driver.trigger.wcdmaMeas.prach.set_threshold(level = 1.0)
```

Defines the trigger threshold for power trigger sources.

param level

No help available

set_timeout(*timeout: float*) → None

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:TOUT
driver.trigger.wcdmaMeas.prach.set_timeout(timeout = 1.0)
```

Selects the maximum time that the CMP180 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

(float or boolean) No help available

Cloning the Group

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

Subgroups

6.3.1.4.1 Catalog

SCPI Command :

```
TRIGger:WCDMa:MEASurement<instance>:PRACH:CATalog:SOURce
```

class CatalogCls

Catalog commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_source() → List[str]

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:PRACH:CATalog:SOURce
value: List[str] = driver.trigger.wcdmaMeas.prach.catalog.get_source()
```

Lists all trigger source values that can be set using method RsCMPX_WcdmaMeas.Trigger.WcdmaMeas.Prach.source.

return

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

6.3.1.5 Tpc

SCPI Commands :

```
TRIGger:WCDma:MEASurement<instance>:TPC:DELay
TRIGger:WCDma:MEASurement<instance>:TPC:MGAP
TRIGger:WCDma:MEASurement<instance>:TPC:THReshold
TRIGger:WCDma:MEASurement<instance>:TPC:SLOPe
TRIGger:WCDma:MEASurement<instance>:TPC:TOUT
TRIGger:WCDma:MEASurement<instance>:TPC:SOURce
```

class TpcCls

Tpc commands group definition. 7 total commands, 1 Subgroups, 6 group commands

get_delay() → float

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:TPC:DELay
value: float = driver.trigger.wcdmaMeas.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 always starts 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: No help available

get_mgap() → float

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:TPC:MGAP
value: float = driver.trigger.wcdmaMeas.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: No help available

get_slope() → SignalSlope

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:TPC:SLOPe
value: enums.SignalSlope = driver.trigger.wcdmaMeas.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: Rising edge FEDGe: Falling edge

get_source() → str

```
# SCPI: TRIGger:WCDma:MEASurement<instance>:TPC:SOURce
value: str = driver.trigger.wcdmaMeas.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: '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.wcdmaMeas.tpc.get_threshold()
```

Defines the trigger threshold for power trigger sources.

return

threshold: No help available

get_timeout() → float

```
# SCPI: TRIGGER:WCDMA:MEASUREMENT<instance>:TPC:TOUT
value: float or bool = driver.trigger.wcdmaMeas.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: (float or boolean) No help available

set_delay(delay: float) → None

```
# SCPI: TRIGGER:WCDMA:MEASUREMENT<instance>:TPC:DELAY
driver.trigger.wcdmaMeas.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 always starts 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

No help available

set_mgap(minimum_gap: float) → None

```
# SCPI: TRIGGER:WCDMA:MEASUREMENT<instance>:TPC:MGAP
driver.trigger.wcdmaMeas.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

No help available

set_slope(*slope*: *SignalSlope*) → None

```
# SCPI: TRIGGER:WCDMA:MEASUREMENT<instance>:TPC:SLOPe
driver.trigger.wcdmaMeas.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: Rising edge FEDGe: Falling edge

set_source(*source*: *str*) → None

```
# SCPI: TRIGGER:WCDMA:MEASUREMENT<instance>:TPC:SOURce
driver.trigger.wcdmaMeas.tpc.set_source(source = 'abc')
```

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

‘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.wcdmaMeas.tpc.set_threshold(threshold = 1.0)
```

Defines the trigger threshold for power trigger sources.

param threshold

No help available

set_timeout(*timeout*: *float*) → None

```
# SCPI: TRIGGER:WCDMA:MEASUREMENT<instance>:TPC:TOUT
driver.trigger.wcdmaMeas.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

(float or boolean) No help available

Cloning the Group

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

Subgroups

6.3.1.5.1 Catalog

SCPI Command :

```
TRIGger:WCDMa:MEASurement<instance>:TPC:CATalog:SOURce
```

class CatalogCls

Catalog commands group definition. 1 total commands, 0 Subgroups, 1 group commands

get_source() → List[str]

```
# SCPI: TRIGger:WCDMa:MEASurement<instance>:TPC:CATalog:SOURce
value: List[str] = driver.trigger.wcdmaMeas.tpc.catalog.get_source()
```

Lists all trigger source values that can be set using method RsCMPX_WcdmaMeas.Trigger.WcdmaMeas.Tpc.source.

return

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

6.4 WcdmaMeas

class WcdmaMeasCls

WcdmaMeas commands group definition. 640 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1 MultiEval

SCPI Commands :

```
STOP:WCDMa:MEASurement<instance>:MEValuation
ABORt:WCDMa:MEASurement<instance>:MEValuation
INITiate:WCDMa:MEASurement<instance>:MEValuation
```

class MultiEvalCls

MultiEval commands group definition. 529 total commands, 8 Subgroups, 3 group commands

abort(opc_timeout_ms: int = -1) → None

```
# SCPI: ABORT:WCDMA:MEASurement<instance>:MEvaluation
driver.wcdmaMeas.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.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

initiate(opc_timeout_ms: int = -1) → None

```
# SCPI: INITiate:WCDMA:MEASurement<instance>:MEvaluation
driver.wcdmaMeas.multiEval.initiate()
```

INTRO_CMD_HELP: Starts, stops **or** aborts the measurement:

- INITiate... starts **or** restarts the measurement. The measurement enters the RUN state.
- STOP... halts the measurement immediately. The measurement enters the RDY state. Measurement results are kept. The resources remain allocated to the measurement.
- ABORT... halts the measurement immediately. The measurement enters the OFF state. All measurement values are **set** to NAV. Allocated resources are released.

Use FETCh...STATe? to query the current measurement state.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

stop(opc_timeout_ms: int = -1) → None

```
# SCPI: STOP:WCDMA:MEASurement<instance>:MEvaluation
driver.wcdmaMeas.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.

(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.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

Cloning the Group

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

Subgroups

6.4.1.1 Ber

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:BER
FETCh:WCDMa:MEASurement<instance>:MEvaluation:BER
```

class BerCls

Ber commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: No parameter help available
- Ber: float: No parameter help available
- Bler: float: No parameter help available

fetch() → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:BER
value: ResultData = driver.wcdmaMeas.multiEval.ber.fetch()
```

No command help available

return

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

read() → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:BER
value: ResultData = driver.wcdmaMeas.multiEval.ber.read()
```

No command help available

return

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

6.4.1.2 Carrier<Carrier>

RepCap Settings

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

class CarrierCls

Carrier commands group definition. 261 total commands, 5 Subgroups, 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.wcdmaMeas.multiEval.carrier.clone()
```

Subgroups

6.4.1.2.1 CdError

class CdErrorCls

CdError commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.1.1 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDERror:AVERage
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDERror:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RMS CDE values for the indicated channels
- Dpdch: float: RMS CDE values for the indicated channels

- Hsdpcch: float: RMS CDE values for the indicated channels
- Edpcch: float: RMS CDE values for the indicated channels
- Edpdch_1: float: RMS CDE values for the indicated channels
- Edpdch_2: float: RMS CDE values for the indicated channels
- Edpdch_3: float: RMS CDE values for the indicated channels
- Edpdch_4: float: RMS CDE values for the indicated channels

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDERror:AVERage
value: ResultData = driver.wcdmaMeas.multiEval.carrier.cdError.average.
↳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*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDERror:AVERage
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.1.2 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDERror:CURRent
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDERror:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RMS CDE values for the indicated channels
- Dpdch: float: RMS CDE values for the indicated channels
- Hsdpcch: float: RMS CDE values for the indicated channels
- Edpcch: float: RMS CDE values for the indicated channels
- Edpdch_1: float: RMS CDE values for the indicated channels
- Edpdch_2: float: RMS CDE values for the indicated channels
- Edpdch_3: float: RMS CDE values for the indicated channels
- Edpdch_4: float: RMS CDE values for the indicated channels

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:CDERror:CURRent
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:CDERror:CURRent
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.1.3 Maximum

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDERror:MAXimum
FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDERror:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RMS CDE values for the indicated channels
- Dpdch: float: RMS CDE values for the indicated channels
- Hsdpcch: float: RMS CDE values for the indicated channels
- Edpcch: float: RMS CDE values for the indicated channels
- Edpdch_1: float: RMS CDE values for the indicated channels
- Edpdch_2: float: RMS CDE values for the indicated channels
- Edpdch_3: float: RMS CDE values for the indicated channels
- Edpdch_4: float: RMS CDE values for the indicated channels

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDERror:MAXimum
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDERror:MAXimum
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.1.4 StandardDev

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDERror:SDEviation
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDERror:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RMS CDE values for the indicated channels
- Dpdch: float: RMS CDE values for the indicated channels
- Hsdpcch: float: RMS CDE values for the indicated channels
- Edpcch: float: RMS CDE values for the indicated channels
- Edpdch_1: float: RMS CDE values for the indicated channels
- Edpdch_2: float: RMS CDE values for the indicated channels
- Edpdch_3: float: RMS CDE values for the indicated channels
- Edpdch_4: float: RMS CDE values for the indicated channels

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:CDERror:SDEviation
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:CDERror:SDEviation
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.2 CdPower

class CdPowerCls

CdPower commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.2.1 Average

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDPower:AVERage
FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDPower:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels
- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels
- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:AVERage
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:AVERage
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.2.2 Current

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDPower:CURRENT
FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDPower:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: ‘Reliability indicator’
- Dpcch: float: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels
- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels

- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(*carrier*=Carrier.Default) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:CDPower:CURRENT
value: ResultData = driver.wcdmaMeas.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) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:CDPower:CURRENT
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.2.3 Maximum

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:CDPower:MAXimum
FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:CDPower:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: ‘Reliability indicator’
- Dpcch: float: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels

- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels
- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:MAXimum
value: ResultData = driver.wcdmaMeas.multiEval.carrier.cdPower.maximum.
↳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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:MAXimum
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.2.4 Minimum

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDPower:MINimum
FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDPower:MINimum
```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels
- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels
- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:MINimum
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:MINimum
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.2.5 StandardDev

SCPI Commands :

```
READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDPower:SDEViation
FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>:CDPower:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels
- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels
- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:SDEViation
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:CDPower:SDEViation
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.3 Modulation

class ModulationCls

Modulation commands group definition. 12 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.3.1 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 AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Evm_Rms: float or bool: Error vector magnitude RMS and peak value.
- Evm_Peak: float or bool: Error vector magnitude RMS and peak value.
- Mag_Error_Rms: float or bool: Magnitude error RMS value.
- Mag_Error_Peak: float or bool: Magnitude error peak value.
- Phase_Error_Rms: float or bool: No parameter help available
- Phase_Error_Peak: float or bool: No parameter help available
- Iq_Offset: float or bool: I/Q origin offset.
- Iq_Imbalance: float or bool: I/Q imbalance.
- Carrier_Freq_Err: float or bool: No parameter help available
- Transmit_Time_Err: float or bool: No parameter help available
- Ue_Power: float or bool: User equipment power.
- Power_Steps: float or bool: User equipment power step.
- Phase_Disc: enums.ResultStatus2: Phase discontinuity.

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Evm_Rms: float: Error vector magnitude RMS and peak value.
- Evm_Peak: float: Error vector magnitude RMS and peak value.
- Mag_Error_Rms: float: Magnitude error RMS value.
- Mag_Error_Peak: float: Magnitude error peak value.
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- Iq_Offset: float: I/Q origin offset.
- Iq_Imbalance: float: I/Q imbalance.
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: User equipment power.
- Power_Steps: float: User equipment power step.
- Phase_Disc: float: Phase discontinuity.
- Tx_Time_Alignment: float: No parameter help available

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:AVERage
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:AVERage
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:MODULATION:AVERAGE
value: ResultData = driver.wcdmaMeas.multiEval.carrier.modulation.average.
↳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.

6.4.1.2.3.2 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 CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: ‘Reliability indicator’
- Evm_Rms: float or bool: Error vector magnitude RMS and peak value.
- Evm_Peak: float or bool: Error vector magnitude RMS and peak value.
- Mag_Error_Rms: float or bool: Magnitude error RMS value.
- Mag_Error_Peak: float or bool: Magnitude error peak value.
- Phase_Error_Rms: float or bool: No parameter help available
- Phase_Error_Peak: float or bool: No parameter help available
- Iq_Offset: float or bool: I/Q origin offset.
- Iq_Imbalance: float or bool: I/Q imbalance.
- Carrier_Freq_Err: float or bool: No parameter help available

- Transmit_Time_Err: float or bool: No parameter help available
- Ue_Power: float or bool: User equipment power.
- Power_Steps: float or bool: User equipment power step.
- Phase_Disc: float or bool: Phase discontinuity.

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Evm_Rms: float: Error vector magnitude RMS and peak value.
- Evm_Peak: float: Error vector magnitude RMS and peak value.
- Mag_Error_Rms: float: Magnitude error RMS value.
- Mag_Error_Peak: float: Magnitude error peak value.
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- Iq_Offset: float: I/Q origin offset.
- Iq_Imbalance: float: I/Q imbalance.
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: User equipment power.
- Power_Steps: float: User equipment power step.
- Phase_Disc: float: Phase discontinuity.
- Tx_Time_Alignment: float: No parameter help available

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:CURRent
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:MODulation:CURRent
value: ResultData = driver.wcdmaMeas.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) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:MODulation:CURRent
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.3.3 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 MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: ‘Reliability indicator’
- Evm_Rms: float or bool: Error vector magnitude RMS and peak value.
- Evm_Peak: float or bool: Error vector magnitude RMS and peak value.
- Mag_Error_Rms: float or bool: Magnitude error RMS value.

- Mag_Error_Peak: float or bool: Magnitude error peak value.
- Phase_Error_Rms: float or bool: No parameter help available
- Phase_Error_Peak: float or bool: No parameter help available
- Iq_Offset: float or bool: I/Q origin offset.
- Iq_Imbalance: float or bool: I/Q imbalance.
- Carrier_Freq_Err: float or bool: No parameter help available
- Transmit_Time_Err: float or bool: No parameter help available
- Ue_Power: float or bool: User equipment power.
- Power_Steps: float or bool: User equipment power step.
- Phase_Disc: enums.ResultStatus2: Phase discontinuity.

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Evm_Rms: float: Error vector magnitude RMS and peak value.
- Evm_Peak: float: Error vector magnitude RMS and peak value.
- Mag_Error_Rms: float: Magnitude error RMS value.
- Mag_Error_Peak: float: Magnitude error peak value.
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- Iq_Offset: float: I/Q origin offset.
- Iq_Imbalance: float: I/Q imbalance.
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: User equipment power.
- Power_Steps: float: User equipment power step.
- Phase_Disc: float: Phase discontinuity.
- Tx_Time_Alignment: float: No parameter help available

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:MAXimum
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:MODulation:MAXimum
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:MODulation:MAXimum
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.3.4 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 StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Evm_Rms: float or bool: Error vector magnitude RMS and peak value.
- Evm_Peak: float or bool: Error vector magnitude RMS and peak value.
- Mag_Error_Rms: float or bool: Magnitude error RMS value.
- Mag_Error_Peak: float or bool: Magnitude error peak value.
- Phase_Error_Rms: float or bool: No parameter help available
- Phase_Error_Peak: float or bool: No parameter help available
- Iq_Offset: float or bool: I/Q origin offset.
- Iq_Imbalance: float or bool: I/Q imbalance.
- Carrier_Freq_Err: float or bool: No parameter help available
- Transmit_Time_Err: float or bool: No parameter help available
- Ue_Power: float or bool: User equipment power.
- Power_Steps: float or bool: User equipment power step.
- Phase_Disc: enums.ResultStatus2: Phase discontinuity.

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Evm_Rms: float: Error vector magnitude RMS and peak value.
- Evm_Peak: float: Error vector magnitude RMS and peak value.
- Mag_Error_Rms: float: Magnitude error RMS value.
- Mag_Error_Peak: float: Magnitude error peak value.
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- Iq_Offset: float: I/Q origin offset.
- Iq_Imbalance: float: I/Q imbalance.
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: User equipment power.
- Power_Steps: float: User equipment power step.
- Phase_Disc: float: Phase discontinuity.
- Tx_Time_Alignment: float: No parameter help available

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:SDEVIation
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:SDEVIation
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:MODulation:SDEVIation
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.4 RcdError

class RcdErrorCls

RcdError commands group definition. 16 total commands, 6 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.4.1 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 AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float or bool: RCDE values for the indicated channels
- Dpdch: float or bool: RCDE values for the indicated channels
- Hsdpcch: float or bool: RCDE values for the indicated channels
- Edpcch: float or bool: RCDE values for the indicated channels
- Edpdch_1: float or bool: RCDE values for the indicated channels
- Edpdch_2: float or bool: RCDE values for the indicated channels
- Edpdch_3: float or bool: RCDE values for the indicated channels
- Edpdch_4: float or bool: RCDE values for the indicated channels

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RCDE values for the indicated channels
- Dpdch: float: RCDE values for the indicated channels
- Hsdpcch: float: RCDE values for the indicated channels
- Edpcch: float: RCDE values for the indicated channels
- Edpdch_1: float: RCDE values for the indicated channels

- Edpdch_2: float: RCDE values for the indicated channels
- Edpdch_3: float: RCDE values for the indicated channels
- Edpdch_4: float: RCDE values for the indicated channels

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:RCDError:AVERage
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:RCDError:AVERage
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:RCDError:AVERage
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.4.2 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 CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float or bool: RCDE values for the indicated channels
- Dpdch: float or bool: RCDE values for the indicated channels
- Hsdpcch: float or bool: RCDE values for the indicated channels
- Edpcch: float or bool: RCDE values for the indicated channels
- Edpdch_1: float or bool: RCDE values for the indicated channels
- Edpdch_2: float or bool: RCDE values for the indicated channels
- Edpdch_3: float or bool: RCDE values for the indicated channels
- Edpdch_4: float or bool: RCDE values for the indicated channels

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RCDE values for the indicated channels
- Dpdch: float: RCDE values for the indicated channels
- Hsdpcch: float: RCDE values for the indicated channels
- Edpcch: float: RCDE values for the indicated channels
- Edpdch_1: float: RCDE values for the indicated channels
- Edpdch_2: float: RCDE values for the indicated channels
- Edpdch_3: float: RCDE values for the indicated channels
- Edpdch_4: float: RCDE values for the indicated channels

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:RCDerror:CURRent
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:RCDerror:CURRENT
value: ResultData = driver.wcdmaMeas.multiEval.carrier.rcdError.current.
↳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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:RCDerror:CURRENT
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.4.3 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 MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float or bool: RCDE values for the indicated channels
- Dpdch: float or bool: RCDE values for the indicated channels
- Hsdpcch: float or bool: RCDE values for the indicated channels
- Edpcch: float or bool: RCDE values for the indicated channels
- Edpdch_1: float or bool: RCDE values for the indicated channels
- Edpdch_2: float or bool: RCDE values for the indicated channels
- Edpdch_3: float or bool: RCDE values for the indicated channels
- Edpdch_4: float or bool: RCDE values for the indicated channels

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RCDE values for the indicated channels
- Dpdch: float: RCDE values for the indicated channels
- Hsdpcch: float: RCDE values for the indicated channels
- Edpcch: float: RCDE values for the indicated channels
- Edpdch_1: float: RCDE values for the indicated channels
- Edpdch_2: float: RCDE values for the indicated channels
- Edpdch_3: float: RCDE values for the indicated channels
- Edpdch_4: float: RCDE values for the indicated channels

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:RCDerror:MAXimum
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:RCDerror:MAXimum
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:RCDerror:MAXimum
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.4.4 OcInfo

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:RCDerror:OCINfo
FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:RCDerror:OCINfo
```

class OcInfoCls

OcInfo commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- State: List[enums.State]: State of the channel OFF: Channel off since start of measurement VAR: The channel has been on and off. ON: Channel on since start of measurement
- Spreading_Factor: List[enums.SpreadingFactorB]: Spreading factor of the channel 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256: constant spreading factor V2 | V4 | V8 | V16 | V32 | V64 | V128 | V256: varying spreading factor, indicates the smallest occurred value
- Modulation: List[enums.Modulation]: Modulation format of the channel BPSK: Constantly BPSK modulated 4PAM: Constantly 4PAM modulated 4PVar: BPSK and 4PAM occurred

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:RCDerror:OCINfo
value: ResultData = driver.wcdmaMeas.multiEval.carrier.rcdError.ocInfo.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the overall channel information for the RCDE measurement. This information is determined from all measured slots.

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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:RCDerror:OCInfo
value: ResultData = driver.wcdmaMeas.multiEval.carrier.rcdError.ocInfo.
↳read(carrier = repcap.Carrier.Default)
```

Returns the overall channel information for the RCDE measurement. This information is determined from all measured slots.

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.

6.4.1.2.4.5 Sf

SCPI Commands :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:RCDerror:Sf
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:RCDerror:Sf
```

class SfCls

Sf commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: int: Spreading factors for the indicated channels
- Dpdch: int: Spreading factors for the indicated channels
- Hsdpcch: int: Spreading factors for the indicated channels
- Edpcch: int: Spreading factors for the indicated channels
- Edpdch_1: int: Spreading factors for the indicated channels
- Edpdch_2: int: Spreading factors for the indicated channels

- Edpdch_3: int: Spreading factors for the indicated channels
- Edpdch_4: int: Spreading factors for the indicated channels

fetch(*carrier=Carrier.Default*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:RCDerror:SF
value: ResultData = driver.wcdmaMeas.multiEval.carrier.rcdError.sf.
↳fetch(carrier = repcap.Carrier.Default)
```

Returns the spreading factors of the dedicated physical channels determined from a selected slot.

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*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:RCDerror:SF
value: ResultData = driver.wcdmaMeas.multiEval.carrier.rcdError.sf.read(carrier,
↳= repcap.Carrier.Default)
```

Returns the spreading factors of the dedicated physical channels determined from a selected slot.

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.

6.4.1.2.4.6 StandardDev

SCPI Commands :

```
CALCulate:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:RCDerror:SDEViation
FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:RCDerror:SDEViation
READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:RCDerror:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: ‘Reliability indicator’
- Dpcch: float or bool: RCDE values for the indicated channels
- Dpdch: float or bool: RCDE values for the indicated channels
- Hsdpcch: float or bool: RCDE values for the indicated channels
- Edpcch: float or bool: RCDE values for the indicated channels

- Edpdch_1: float or bool: RCDE values for the indicated channels
- Edpdch_2: float or bool: RCDE values for the indicated channels
- Edpdch_3: float or bool: RCDE values for the indicated channels
- Edpdch_4: float or bool: RCDE values for the indicated channels

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Dpcch: float: RCDE values for the indicated channels
- Dpdch: float: RCDE values for the indicated channels
- Hsdpcch: float: RCDE values for the indicated channels
- Edpcch: float: RCDE values for the indicated channels
- Edpdch_1: float: RCDE values for the indicated channels
- Edpdch_2: float: RCDE values for the indicated channels
- Edpdch_3: float: RCDE values for the indicated channels
- Edpdch_4: float: RCDE values for the indicated channels

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:RCDError:SDEViation
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:RCDError:SDEViation
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:RCDError:SDEVIation
value: ResultData = driver.wcdmaMeas.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.

6.4.1.2.5 Trace**class TraceCls**

Trace commands group definition. 215 total commands, 9 Subgroups, 0 group commands

Cloning the Group

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

Subgroups**6.4.1.2.5.1 CdError****class CdErrorCls**

CdError commands group definition. 40 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.5.2 Dpcch

class DpcchCls

Dpcch commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.cdError.dpcch.clone()
```

Subgroups

6.4.1.2.5.3 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:CDERror:DPCCh:AVERage
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDERror:DPCCh:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDERror:DPCCh:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.dpcch.
↳average.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:DPCCh:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.4 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:DPCCh:CURREnt
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:CDError:DPCCh:CURRENT
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.dpcch.
↳ current.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.5 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:CDError:DPCCh:MAXIMUM
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDError:DPCCh:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.6 StandardDev

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDError:DPCCh:SDEviation
FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDError:DPCCh:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDError:DPCCh:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDError:DPCh:SDEViation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.7 Dpdch

class DpdchCls

Dpdch commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.cdError.dpdch.clone()
```

Subgroups

6.4.1.2.5.8 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDError:DPDCh:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.dpdch.
↳average.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDError:DPDCh:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.dpdch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.9 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier*=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:CDError:DPDCh:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier*=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:CDError:DPDCh:CURRENT
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.dpdch.
↳current.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.10 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDError:DPDCh:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.dpdch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDError:DPDCh:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.11 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:DPDCh:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.dpdch.
↳standardDev.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:DPDCh:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.12 Edpcch

class EdpcchCls

Edpcch commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.cdError.edpcch.clone()
```

Subgroups

6.4.1.2.5.13 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.14 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:CURRent
```

(continues on next page)

(continued from previous page)

```
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.15 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>  
↳:TRACe:CDError:EDPCch:MAXimum  
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:CDError:EDPCch:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.16 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:CDError:EDPCch:SDEVIATION
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPCch:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.17 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.wcdmaMeas.multiEval.carrier.trace.cdError.edpdch.repcap_edpdChannel_get()
driver.wcdmaMeas.multiEval.carrier.trace.cdError.edpdch.repcap_edpdChannel_set(repcap.
↳EdpdChannel.Nr1)
```

class EdpdchCls

Edpdch commands group definition. 8 total commands, 4 Subgroups, 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.wcdmaMeas.multiEval.carrier.trace.cdError.edpdch.clone()
```

Subgroups

6.4.1.2.5.18 Average

SCPI Commands :

```
READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:CDERror:EDPDch<nr>
↳:AVERage
FETCH:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:CDERror:EDPDch<nr>
↳:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:EDPDch<nr>:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDE trace results, one result per measured slot or half-slot

read(*carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:EDPDch<nr>:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDE trace results, one result per measured slot or half-slot

6.4.1.2.5.19 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:CDERror:EDPDch<nr>
↳:CURRent
FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:CDERror:EDPDch<nr>
↳:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDERror:EDPDch<nr>:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDE trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDErRor:EDPDch<nr>:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDE trace results, one result per measured slot or half-slot

6.4.1.2.5.20 Maximum

SCPI Commands :

```
READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:CDErRor:EDPDch<nr>
↳:MAXimum
FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:CDErRor:EDPDch<nr>
↳:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default, edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDErRor:EDPDch<nr>:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDE trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPDch<nr>:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDE trace results, one result per measured slot or half-slot

6.4.1.2.5.21 StandardDev

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:CDError:EDPDch<nr>
↳:SDEviation
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:CDError:EDPDch<nr>
↳:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:EDPDch<nr>:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.edpdch.
```

(continues on next page)

(continued from previous page)

```
↪ 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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDE trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↪ :TRACe:CDError:EDPDch<nr>:SDEVIation
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDE trace results, one result per measured slot or half-slot

6.4.1.2.5.22 Hsdpcch

class HsdpcchCls

Hsdpcch commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.cdError.hsdpcch.clone()
```

Subgroups

6.4.1.2.5.23 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.hsdpcch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.24 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:CURRENT
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.hsdpcch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:CURRENT
```

(continues on next page)

(continued from previous page)

```
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.25 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>  
↳:TRACe:CDError:HSDPcch:MAXimum  
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.hsdpcch.  
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↪:TRACE:CDError:HSDPCCH:MAXIMUM
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.26 StandardDev

SCPI Commands :

```
READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↪:TRACE:CDError:HSDPCCH:SDEVIACTION
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↪:TRACE:CDError:HSDPCCH:SDEVIACTION
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↪:TRACE:CDError:HSDPCCH:SDEVIACTION
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdError.hsdpcch.
↪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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: RMS CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDError:HSDPcch:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: RMS CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.27 CdPower

class CdPowerCls

CdPower commands group definition. 50 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.5.28 Dpcch

class DpcchCls

Dpcch commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.dpcch.clone()
```

Subgroups

6.4.1.2.5.29 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPCCh:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPCCh:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.30 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:DPCCh:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:DPCCh:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.31 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:DPCCh:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:DPCCh:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.32 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 MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:DPCCh:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:DPCCh:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.33 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:DPCCh:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:DPCCh:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.dpcch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of

results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.34 Dpdch

class DpdchCls

Dpdch commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.dpdch.clone()
```

Subgroups

6.4.1.2.5.35 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.dpdch.
↳average.fetch(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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:CDPOWER:DPDCH:AVERAGE
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.36 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:CDPOWER:DPDCH:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:CDPOWER:DPDCH:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.37 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:CDPOWER:DPDCH:MAXIMUM
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:CDPOWER:DPDCH:MAXIMUM
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.38 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 MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:CDPOWER:DPDCH:MINIMUM
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.39 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:DPDCh:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.40 Edpcch

class EdpcchCls

Edpcch commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.edpcch.clone()
```

Subgroups

6.4.1.2.5.41 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier*=*Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:EDPCch:AVERage
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation`) . The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

`edpcch`: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier*=*Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:CDPower:EDPCch:AVERage
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation`) . The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

`edpcch`: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.42 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```

# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```

# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.43 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.44 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 MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.45 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:EDPCch:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.46 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.edpdch.repcap_edpdChannel_get()
driver.wcdmaMeas.multiEval.carrier.trace.cdPower.edpdch.repcap_edpdChannel_set(repcap.
↳ EdpdChannel.Nr1)
```

class EdpdchCls

Edpdch commands group definition. 10 total commands, 5 Subgroups, 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.wcdmaMeas.multiEval.carrier.trace.cdPower.edpdch.clone()
```

Subgroups

6.4.1.2.5.47 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:CDPower:EDPDch<nr>
↳ :AVERage
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:CDPower:EDPDch<nr>
↳ :AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default, edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳ :TRACe:CDPower:EDPDch<nr>:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas. 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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:EDPDch<nr>:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.48 Current

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:CDPower:EDPDch<nr>
↳:CURRENT
FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:CDPower:EDPDch<nr>
↳:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:EDPDch<nr>:CURRENT
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳ :TRACe:CDPower:EDPDch<nr>:CURRent
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.49 Maximum

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:CDPower:EDPDch<nr>
↳:MAXimum
FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:CDPower:EDPDch<nr>
↳:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas. 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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

read(*carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:CDPower:EDPDch<nr>:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas. 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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.50 Minimum**SCPI Commands :**

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACe:CDPower:EDPDch<nr>
↳:MINimum
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACe:CDPower:EDPDch<nr>
↳:MINimum
```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default, edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACe:CDPower:EDPDch<nr>:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas. 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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default, edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACe:CDPower:EDPDch<nr>:MINimum
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.edpdch.
```

(continues on next page)

(continued from previous page)

```
↪ 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas. 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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.51 StandardDev

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:CDPower:EDPDch<nr>
↪:SDEviation
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:CDPower:EDPDch<nr>
↪:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default, edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:CDPower:EDPDch<nr>:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas. 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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default, edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:CDPOWER:EDPDCH<nr>:SDEViation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

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: RMS CDP trace results, one result per measured slot or half-slot

6.4.1.2.5.52 Hsdpcch**class HsdpcchCls**

Hsdpcch commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.hsdpcch.clone()
```

Subgroups

6.4.1.2.5.53 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.54 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.hsdpccch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.hsdpccch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.55 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.hsdpccch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.cdPower.hsdpccch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.56 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 MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:HSDPcch:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:CDPower:HSDPcch:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. Wcd-

maMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.57 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: RMS CDP trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:CDPower:HSDPcch:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcc: RMS CDP trace results, one result per measured slot or half-slot.

6.4.1.2.5.58 EvMagnitude

class EvMagnitudeCls

EvMagnitude commands group definition. 16 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.5.59 Peak

class PeakCls

Peak commands group definition. 8 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.evMagnitude.peak.clone()
```

Subgroups

6.4.1.2.5.60 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: Peak EVM trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: Peak EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.61 Current

SCPI Commands :

```

FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:VMagnitude:PEAK:CURRent
READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:VMagnitude:PEAK:CURRent

```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```

# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:VMagnitude:PEAK:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: Peak EVM trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```

# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:VMagnitude:PEAK:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: Peak EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.62 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```

# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: Peak EVM trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```

# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: Peak EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.63 Sdeviaton

SCPI Command :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:SDEVIaton
```

class SdeviatonCls

Sdeviaton commands group definition. 1 total commands, 0 Subgroups, 1 group commands

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:SDEVIaton
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: Peak EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.64 StandardDev

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:SDEVIation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude:PEAK:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.

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.

Suppressed linked return values: reliability

param carrier

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

return

evm: Peak EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.65 Rms

class RmsCls

Rms commands group definition. 8 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.evMagnitude.rms.clone()
```

Subgroups

6.4.1.2.5.66 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: RMS EVM trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: RMS EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.67 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: RMS EVM trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: RMS EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.68 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.

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.

Suppressed linked return values: reliability

param carrier

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

return

evm: RMS EVM trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: RMS EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.69 Sdeviaton

SCPI Command :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:SDEViaton
```

class SdeviatonCls

Sdeviaton commands group definition. 1 total commands, 0 Subgroups, 1 group commands

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:EVMagnitude[:RMS]:SDEViaton
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.

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.

Suppressed linked return values: reliability

param carrier

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

return

evm: RMS EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.70 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:EVMAgnitude[:RMS]:SDEVIATION
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:EVMAgnitude[:RMS]:SDEVIATION
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

evm: RMS EVM trace results, one result per measured slot or half-slot

6.4.1.2.5.71 FreqError

class FreqErrorCls

FreqError commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.5.72 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:FERRor:AVERage
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:FERRor:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:FERRor:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

frequency_error: Carrier frequency error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:FERRor:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

frequency_error: Carrier frequency error trace results, one result per measured slot or half-slot

6.4.1.2.5.73 Current

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:FERRor:CURRent
FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:FERRor:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:FERRor:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

frequency_error: Carrier frequency error trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:FERRor:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of

results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

frequency_error: Carrier frequency error trace results, one result per measured slot or half-slot

6.4.1.2.5.74 Maximum

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:FERRor:MAXimum
FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:FERRor:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:FERRor:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

frequency_error: Carrier frequency error trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:FERRor:MAXimum
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

frequency_error: Carrier frequency error trace results, one result per measured slot or half-slot

6.4.1.2.5.75 StandardDev

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:FERRor:SDEViation
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:FERRor:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:FERRor:SDEViation
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

frequency_error: Carrier frequency error trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:FERRor:SDEViation
```

(continues on next page)

(continued from previous page)

```
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

frequency_error: Carrier frequency error trace results, one result per measured slot or half-slot

6.4.1.2.5.76 Merror

class MerrorCls

Merror commands group definition. 16 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.5.77 Peak

class PeakCls

Peak commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently  
group2 = driver.wcdmaMeas.multiEval.carrier.trace.merror.peak.clone()
```

Subgroups

6.4.1.2.5.78 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```

# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
→ :TRACe:MERRor:PEAK:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: Peak magnitude error trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```

# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
→ :TRACe:MERRor:PEAK:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: Peak magnitude error trace results, one result per measured slot or half-slot

6.4.1.2.5.79 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR:PEAK:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: Peak magnitude error trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR:PEAK:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: Peak magnitude error trace results, one result per measured slot or half-slot

6.4.1.2.5.80 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:MERRor:PEAK:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: Peak magnitude error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:MERRor:PEAK:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: Peak magnitude error trace results, one result per measured slot or half-slot

6.4.1.2.5.81 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```

# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:MERRor:PEAK:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: Peak magnitude error trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```

# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:MERRor:PEAK:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of

results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: Peak magnitude error trace results, one result per measured slot or half-slot

6.4.1.2.5.82 Rms

class RmsCls

Rms commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.merror.rms.clone()
```

Subgroups

6.4.1.2.5.83 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRACe:MERRor[:RMS]:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: RMS magnitude error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR[:RMS]:AVERAGE
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: RMS magnitude error trace results, one result per measured slot or half-slot

6.4.1.2.5.84 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR[:RMS]:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: RMS magnitude error trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:MERRor[:RMS]:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: RMS magnitude error trace results, one result per measured slot or half-slot

6.4.1.2.5.85 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:MERRor[:RMS]:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.

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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: RMS magnitude error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR[:RMS]:MAXIMUM
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: RMS magnitude error trace results, one result per measured slot or half-slot

6.4.1.2.5.86 StandardDev

SCPI Commands :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR[:RMS]:SDEVATION
READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR[:RMS]:SDEVATION
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR[:RMS]:SDEVATION
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: RMS magnitude error trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:MERROR[:RMS]:SDEVIATION
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

magnitude_error: RMS magnitude error trace results, one result per measured slot or half-slot

6.4.1.2.5.87 Perror

class PerrorCls

Perror commands group definition. 15 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.5.88 Peak

class PeakCls

Peak commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.perror.peak.clone()
```

Subgroups

6.4.1.2.5.89 Average

SCPI Commands :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:PERRor:PEAK:AVERage
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:PERRor:PEAK:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:PERRor:PEAK:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: Peak phase error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:PERROR:PEAK:AVERAGE
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: Peak phase error trace results, one result per measured slot or half-slot

6.4.1.2.5.90 Current

SCPI Commands :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>:TRACE:PERROR:PEAK:CURRENT
READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>:TRACE:PERROR:PEAK:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:PERROR:PEAK:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: Peak phase error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:PERROR:PEAK:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: Peak phase error trace results, one result per measured slot or half-slot

6.4.1.2.5.91 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳ :TRACE:PERROR:PEAK:MAXIMUM
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: Peak phase error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:PERRor:PEAK:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: Peak phase error trace results, one result per measured slot or half-slot

6.4.1.2.5.92 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:PERRor:PEAK:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: Peak phase error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:PERRor:PEAK:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: Peak phase error trace results, one result per measured slot or half-slot

6.4.1.2.5.93 Rms

class RmsCls

Rms commands group definition. 7 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.perror.rms.clone()
```

Subgroups

6.4.1.2.5.94 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier*=*Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:PERRor[:RMS]:AVERage
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

`phase_error`: RMS phase error trace results, one result per measured slot or half-slot

read(*carrier*=*Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:PERRor[:RMS]:AVERage
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

`phase_error`: RMS phase error trace results, one result per measured slot or half-slot

6.4.1.2.5.95 Current

SCPI Command :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>:TRACE:PERROR[:RMS]:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
→ :TRACE:PERROR[:RMS]:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: RMS phase error trace results, one result per measured slot or half-slot

6.4.1.2.5.96 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
→ :TRACE:PERROR[:RMS]:MAXIMUM
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.

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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: RMS phase error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:PERRor[:RMS]:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: RMS phase error trace results, one result per measured slot or half-slot

6.4.1.2.5.97 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:PERRor[:RMS]:SDEviation
value: List[float] = driver.wcdmaMeas.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

RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: RMS phase error trace results, one result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪:TRAcE:PERRor[:RMS]:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: RMS phase error trace results, one result per measured slot or half-slot

6.4.1.2.5.98 Psteps

class PstepsCls

Psteps commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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


Subgroups

6.4.1.2.5.99 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:PSTeps:AVERage
FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:PSTeps:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:PSTeps:AVERage
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:PSTeps:AVERage
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

6.4.1.2.5.100 Current**SCPI Commands :**

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRAcE:PSTeps:CURRent
FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRAcE:PSTeps:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRAcE:PSTeps:CURRent
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRAcE:PSTeps:CURRent
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

6.4.1.2.5.101 Maximum

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:PSTeps:MAXimum
FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:PSTeps:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:PSTeps:MAXimum
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:PSTeps:MAXimum
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

6.4.1.2.5.102 Minimum

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:PSTeps:MINimum
FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:PSTeps:MINimum
```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↪:TRACe:PSTeps:MINimum
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:PSTeps:MINimum
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

6.4.1.2.5.103 StandardDev

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:PSTeps:SDEViation
FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:PSTeps:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:PSTeps:SDEViation
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACe:PSTeps:SDEVIation
value: List[float] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: One result per measured slot or halfslot

6.4.1.2.5.104 RcdError**class RcdErrorCls**

RcdError commands group definition. 50 total commands, 6 Subgroups, 0 group commands

Cloning the Group

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

Subgroups**6.4.1.2.5.105 Dpcch****class DpcchCls**

Dpcch commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.dpcch.clone()
```

Subgroups

6.4.1.2.5.106 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. Wcd-

maMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.107 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.108 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.109 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```

# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:SDEViation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```

# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPCCh:SDEViation

```

(continues on next page)

(continued from previous page)

```
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.110 Dpdch

class DpdchCls

Dpdch commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently  
group2 = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.dpdch.clone()
```

Subgroups

6.4.1.2.5.111 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>  
↪ :TRACe:RCDerror:DPDCh:AVERage  
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.112 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:DPDCh:CURREnt
```

(continues on next page)

(continued from previous page)

```
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Relative CDE trace results, one result per measured slot or half-slot.

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>  
↪:TRACe:RCDerror:DPDCh:CURRent  
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.113 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier*=Carrier.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDerror:DPDCh:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.dpdch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier*=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDerror:DPDCh:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.dpdch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.114 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:RCDError:DPDCh:SDEVIation
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.dpdch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACE:RCDError:DPDCh:SDEVIation
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.dpdch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.115 Edpcch

class EdpcchCls

Edpcch commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.edpcch.clone()
```

Subgroups

6.4.1.2.5.116 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:EDPCch:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:EDPCch:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.117 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:EDPCch:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:EDPCch:CURRent
```

(continues on next page)

(continued from previous page)

```
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.118 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>  
↪:TRACe:RCDerror:EDPCch:MAXimum  
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDerror:EDPCch:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.119 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 StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDerror:EDPCch:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDError:EDPCch:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.120 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.edpdch.repcap_edpdChannel_get()
driver.wcdmaMeas.multiEval.carrier.trace.rcdError.edpdch.repcap_edpdChannel_set(repcap.
↳EdpdChannel.Nr1)
```

class EdpdchCls

Edpdch commands group definition. 8 total commands, 4 Subgroups, 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.wcdmaMeas.multiEval.carrier.trace.rcdError.edpdch.clone()
```

Subgroups

6.4.1.2.5.121 Average

SCPI Commands :

```
READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:RCDerror:EDPDch<nr>
↳:AVERage
FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:RCDerror:EDPDch<nr>
↳:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

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: Relative CDE trace results, one result per measured slot or half-slot

read(*carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

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: Relative CDE trace results, one result per measured slot or half-slot

6.4.1.2.5.122 Current

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:RCDerror:EDPDch<nr>
↳:CURRENT
FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:RCDerror:EDPDch<nr>
↳:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDerror:EDPDch<nr>:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

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: Relative CDE trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

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: Relative CDE trace results, one result per measured slot or half-slot

6.4.1.2.5.123 Maximum

SCPI Commands :

```
READ:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:RCDerror:EDPDch<nr>
↳:MAXimum
FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:RCDerror:EDPDch<nr>
↳:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default, edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:EDPDch<nr>:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

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: Relative CDE trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDError:EDPDch<nr>:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

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: Relative CDE trace results, one result per measured slot or half-slot

6.4.1.2.5.124 StandardDev

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:RCDError:EDPDch<nr>
↳:SDEviation
FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:RCDError:EDPDch<nr>
↳:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDError:EDPDch<nr>:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.edpdch.
```

(continues on next page)

(continued from previous page)

```
↪ 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

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: Relative CDE trace results, one result per measured slot or half-slot

read(*carrier=Carrier.Default, edpdChannel=EdpdChannel.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↪ :TRACe:RCDeRRor:EDPDch<nr>:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval. Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.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.

Suppressed linked return values: reliability

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: Relative CDE trace results, one result per measured slot or half-slot

6.4.1.2.5.125 Hsdpcch

class HsdpcchCls

Hsdpcch commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.hsdpcch.clone()
```

Subgroups

6.4.1.2.5.126 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 AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:HSDPcch:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:HSDPcch:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.127 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 CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:HSDPcch:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.hsdpccch.
↳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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpccch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:HSDPcch:CURRent
```

(continues on next page)

(continued from previous page)

```
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.128 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 MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>  
↪:TRACe:RCDerror:HSDPcch:MAXimum  
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:RCDERROR:HSDPCCH:MAXIMUM
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.129 StandardDev

SCPI Commands :

```
READ:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:RCDERROR:HSDPCCH:SDEVIACTION
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:RCDERROR:HSDPCCH:SDEVIACTION
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:CARRIER<carrier>
↳:TRACE:RCDERROR:HSDPCCH:SDEVIACTION
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: Relative CDE trace results, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:HSDPcch:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas. MultiEval.Mperiod.modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: Relative CDE trace results, one result per measured slot or half-slot.

6.4.1.2.5.130 Sf

class SfCls

Sf commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.sf.clone()
```

Subgroups

6.4.1.2.5.131 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 DpcchCls

Dpcch commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[int]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:SF:DPCCh
value: List[int] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount`).

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Spreading factors, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[int]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:SF:DPCCh
value: List[int] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount`).

Suppressed linked return values: reliability

param carrier

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

return

dpcch: Spreading factors, one result per measured slot or half-slot.

6.4.1.2.5.132 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 DpdchCls

Dpdch commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[int]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:SF:DPDCh
value: List[int] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount`).

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Spreading factors, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[int]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:RCDerror:SF:DPDCh
value: List[int] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount`).

Suppressed linked return values: reliability

param carrier

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

return

dpdch: Spreading factors, one result per measured slot or half-slot.

6.4.1.2.5.133 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 EdpcchCls

Edpcch commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[int]


```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:SF:EDPCch
value: List[int] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount`).

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Spreading factors, one result per measured slot or half-slot.

read(*carrier=Carrier.Default*) → List[int]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:SF:EDPCch
value: List[int] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount`).

Suppressed linked return values: reliability

param carrier

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

return

edpcch: Spreading factors, one result per measured slot or half-slot.

6.4.1.2.5.134 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.sf.edpdch.repcap_edpdChannel_get()
driver.wcdmaMeas.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
↳<nr>
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:RCDerror:SF:EDPDch
↳<nr>

```

class EdpdchCls

Edpdch commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: EdpdChannel, default value after init: EdpdChannel.Nr1

fetch(*carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → List[float]

```

# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDerror:SF:EDPDch<nr>
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount).

Suppressed linked return values: reliability

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: Spreading factors, one result per measured slot or half-slot

read(*carrier*=Carrier.Default, *edpdChannel*=EdpdChannel.Default) → List[float]

```

# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:RCDerror:SF:EDPDch<nr>
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount).

Suppressed linked return values: reliability

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: Spreading factors, one result per measured slot or half-slot

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.carrier.trace.rcdError.sf.edpdch.clone()
```

6.4.1.2.5.135 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 HsdpcchCls

Hsdpcch commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[int]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:SF:HSDPcch
value: List[int] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`). The number of results depends on the measurement length (see method `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount`).

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: Spreading factors, one result per measured slot or half-slot.

read(carrier=Carrier.Default) → List[int]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:RCDerror:SF:HSDPcch
value: List[int] = driver.wcdmaMeas.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 `RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod.modulation`).

. The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount) .

Suppressed linked return values: reliability

param carrier

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

return

hsdpcch: Spreading factors, one result per measured slot or half-slot.

6.4.1.2.5.136 UePower

class UePowerCls

UePower commands group definition. 10 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.2.5.137 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:UEPower:AVERage
FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:UEPower:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳ :TRACe:UEPower:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas. 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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:UEPower:AVERage
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

6.4.1.2.5.138 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:UEPower:CURRENT
FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>:TRACe:UEPower:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(carrier=Carrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRier<carrier>
↳:TRACe:UEPower:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:UEPower:CURRENT
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

6.4.1.2.5.139 Maximum

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:UEPower:MAXimum
FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:UEPower:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:UEPower:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:UEPower:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

6.4.1.2.5.140 Minimum

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:UEPower:MINimum
FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>:TRACe:UEPower:MINimum
```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRier<carrier>
↳:TRACe:UEPower:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:UEPower:MINimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

6.4.1.2.5.141 StandardDev

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:UEPower:SDEviation
FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>:TRACE:UEPower:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACE:UEPower:SDEviation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:CARRIER<carrier>
↳:TRACe:UEPower:SDEVIation
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation). The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: One result per measured slot or half-slot

6.4.1.3 ListPy

class ListPyCls

ListPy commands group definition. 181 total commands, 9 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.1 CdError

class CdErrorCls

CdError commands group definition. 24 total commands, 9 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.1.1 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: RMS CDE values for the indicated channels
- Dpdch: List[float]: RMS CDE values for the indicated channels
- Hsdpcch: List[float]: RMS CDE values for the indicated channels
- Edpcch: List[float]: RMS CDE values for the indicated channels
- Edpdch_1: List[float]: RMS CDE values for the indicated channels
- Edpdch_2: List[float]: RMS CDE values for the indicated channels
- Edpdch_3: List[float]: RMS CDE values for the indicated channels
- Edpdch_4: List[float]: RMS CDE values for the indicated channels

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:AVERage
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.1.2 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: RMS CDE values for the indicated channels
- Dpdch: List[float]: RMS CDE values for the indicated channels
- Hsdpcch: List[float]: RMS CDE values for the indicated channels
- Edpcch: List[float]: RMS CDE values for the indicated channels
- Edpdch_1: List[float]: RMS CDE values for the indicated channels
- Edpdch_2: List[float]: RMS CDE values for the indicated channels
- Edpdch_3: List[float]: RMS CDE values for the indicated channels
- Edpdch_4: List[float]: RMS CDE values for the indicated channels

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:CURRent
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.1.3 Dpcch

class DpcchCls

Dpcch commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.1.4 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.dpcch.average.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

dpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.5 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.dpcch.current.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

dpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.6 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.dpcch.maximum.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.7 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPCCh:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪ :MEvaluation:LIST:CDError:DPCCh:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.dpcch.
↪ standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.8 Dpdch

class DpdchCls

Dpdch commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.1.9 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.dpdch.average.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Suppressed linked return values: reliability

return

dpdch: Comma-separated list of values, one per measured segment

6.4.1.3.1.10 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:CURREnt
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:CURREnt
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.dpdch.current.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Suppressed linked return values: reliability

return

dpdch: Comma-separated list of values, one per measured segment

6.4.1.3.1.11 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.dpdch.maximum.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Suppressed linked return values: reliability

return

dpdch: Comma-separated list of values, one per measured segment

6.4.1.3.1.12 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:DPDCh:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪ :MEvaluation:LIST:CDError:DPDCh:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.dpdch.
↪ standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Suppressed linked return values: reliability

return

dpdch: Comma-separated list of values, one per measured segment

6.4.1.3.1.13 Edpcch

class EdpcchCls

Edpcch commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.1.14 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:EDPCch:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:CDError:EDPCch:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.edpcch.average.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

edpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.15 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDError:EDPCch:CURREnt
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:CDError:EDPCch:CURREnt
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.edpcch.current.
↳fetch()
```


Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
edpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.16 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDERror:EDPCch:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:CDERror:EDPCch:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.edpcch.maximum.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
edpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.17 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDERror:EDPCch:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:CDERror:EDPCch:SDEViation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.edpcch.
↳standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
edpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.18 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.wcdmaMeas.multiEval.listPy.cdError.edpdch.repcap_edpdChannel_get()
driver.wcdmaMeas.multiEval.listPy.cdError.edpdch.repcap_edpdChannel_set(repcap.
↳ EdpdChannel.Nr1)
```

class EdpdchCls

Edpdch commands group definition. 4 total commands, 4 Subgroups, 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.wcdmaMeas.multiEval.listPy.cdError.edpdch.clone()
```

Subgroups

6.4.1.3.1.19 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:EDPDch<nr>:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:EDPDch<nr>
↳ :AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param edpdChannel

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

return

edpdch: Comma-separated list of values, one per measured segment

6.4.1.3.1.20 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:EDPDch<nr>:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:EDPDch<nr>
↪:CURRENT
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param edpdChannel

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

return

edpdch: Comma-separated list of values, one per measured segment

6.4.1.3.1.21 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:EDPDch<nr>:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:EDPDch<nr>
↪:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param edpdChannel

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

return

edpdch: Comma-separated list of values, one per measured segment

6.4.1.3.1.22 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:EDPDch<nr>:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:EDPDch<nr>
↳:SDEviation
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param edpdChannel

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

return

edpdch: Comma-separated list of values, one per measured segment

6.4.1.3.1.23 Hsdpcch

class HsdpcchCls

Hsdpcch commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.1.24 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:HSDPcch:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>
↳:MEVALUATION:LIST:CDError:HSDPCCH:AVERAge
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.hsdpcch.average.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

hsdpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.25 Current

SCPI Command :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:CDError:HSDPCCH:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>
↳:MEVALUATION:LIST:CDError:HSDPCCH:CURRENT
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.hsdpcch.current.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

hsdpcch: Comma-separated list of values, one per measured segment

6.4.1.3.1.26 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:CDError:HSDPCCH:MAXIMUM
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>
↳:MEVALUATION:LIST:CDError:HSDPCCH:MAXIMUM
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.hsdpcch.maximum.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

hsdpccch: Comma-separated list of values, one per measured segment

6.4.1.3.1.27 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:HSDPcch:SDEVIation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:CDError:HSDPcch:SDEVIation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdError.hsdpccch.
↳standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

hsdpccch: Comma-separated list of values, one per measured segment

6.4.1.3.1.28 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: RMS CDE values for the indicated channels
- Dpdch: List[float]: RMS CDE values for the indicated channels
- Hsdpccch: List[float]: RMS CDE values for the indicated channels
- Edpcch: List[float]: RMS CDE values for the indicated channels
- Edpdch_1: List[float]: RMS CDE values for the indicated channels
- Edpdch_2: List[float]: RMS CDE values for the indicated channels

- Edpdch_3: List[float]: RMS CDE values for the indicated channels
- Edpdch_4: List[float]: RMS CDE values for the indicated channels

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:MAXimum
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.1.29 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: RMS CDE values for the indicated channels
- Dpdch: List[float]: RMS CDE values for the indicated channels
- Hsdpcch: List[float]: RMS CDE values for the indicated channels
- Edpcch: List[float]: RMS CDE values for the indicated channels
- Edpdch_1: List[float]: RMS CDE values for the indicated channels
- Edpdch_2: List[float]: RMS CDE values for the indicated channels
- Edpdch_3: List[float]: RMS CDE values for the indicated channels
- Edpdch_4: List[float]: RMS CDE values for the indicated channels

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDError:SDEviation
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.2 CdPower

class CdPowerCls

CdPower commands group definition. 30 total commands, 10 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.2.1 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDPower:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: RMS CDP values for the indicated channels
- Dpdch: List[float]: RMS CDP values for the indicated channels
- Hsdpcch: List[float]: RMS CDP values for the indicated channels
- Edpcch: List[float]: RMS CDP values for the indicated channels
- Edpdch_1: List[float]: RMS CDP values for the indicated channels
- Edpdch_2: List[float]: RMS CDP values for the indicated channels
- Edpdch_3: List[float]: RMS CDP values for the indicated channels
- Edpdch_4: List[float]: RMS CDP values for the indicated channels

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDPower:AVERage
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.2.2 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: RMS CDP values for the indicated channels
- Dpdch: List[float]: RMS CDP values for the indicated channels
- Hsdpcch: List[float]: RMS CDP values for the indicated channels
- Edpcch: List[float]: RMS CDP values for the indicated channels
- Edpdch_1: List[float]: RMS CDP values for the indicated channels
- Edpdch_2: List[float]: RMS CDP values for the indicated channels
- Edpdch_3: List[float]: RMS CDP values for the indicated channels
- Edpdch_4: List[float]: RMS CDP values for the indicated channels

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:CURRent
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.2.3 Dpcch

class DpcchCls

Dpcch commands group definition. 5 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.2.4 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:DPCCh:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:DPCCh:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpcch.average.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

dpcch: Comma-separated list of values, one per measured segment

6.4.1.3.2.5 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:DPCCh:CURREnt
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:DPCCh:CURREnt
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpcch.current.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpcch: Comma-separated list of values, one per measured segment

6.4.1.3.2.6 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPCCh:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPCCh:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpcch.maximum.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpcch: Comma-separated list of values, one per measured segment

6.4.1.3.2.7 Minimum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPCCh:MINimum
```

class MinimumCls

Minimum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPCCh:MINimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpcch.minimum.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpcch: Comma-separated list of values, one per measured segment

6.4.1.3.2.8 StandardDev

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDPower:DPCCh:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:CDPower:DPCCh:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpcch.
↳standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpcch: Comma-separated list of values, one per measured segment

6.4.1.3.2.9 Dpdch

class DpdchCls

Dpdch commands group definition. 5 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.2.10 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpdch.average.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.11 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpdch.current.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.12 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpdch.maximum.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.13 Minimum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:MINimum
```

class MinimumCls

Minimum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:MINimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpdch.minimum.
↳ fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.14 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:DPDCh:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳ :MEvaluation:LIST:CDPower:DPDCh:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.dpdch.
↳ standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the DPDCH for all measured list mode segments.

Suppressed linked return values: reliability

return
dpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.15 Edpcch

class EdpcchCls

Edpcch commands group definition. 5 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.2.16 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPCch:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪ :MEvaluation:LIST:CDPower:EDPCch:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.edpcch.average.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

edpcch: Comma-separated list of values, one per measured segment

6.4.1.3.2.17 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPCch:CURREnt
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪ :MEvaluation:LIST:CDPower:EDPCch:CURREnt
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.edpcch.current.
↪ fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    edpcch: Comma-separated list of values, one per measured segment
```

6.4.1.3.2.18 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPCch:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:CDPower:EDPCch:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.edpcch.maximum.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    edpcch: Comma-separated list of values, one per measured segment
```

6.4.1.3.2.19 Minimum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPCch:MINimum
```

class MinimumCls

Minimum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:CDPower:EDPCch:MINimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.edpcch.minimum.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    edpcch: Comma-separated list of values, one per measured segment
```


6.4.1.3.2.20 StandardDev

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPCch:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:CDPower:EDPCch:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.edpcch.
↪standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the E-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

edpcch: Comma-separated list of values, one per measured segment

6.4.1.3.2.21 Edpdch<EdpdChannel>

RepCap Settings

```
# Range: Nr1 .. Nr4
rc = driver.wcdmaMeas.multiEval.listPy.cdPower.edpdch.repcap_edpdChannel_get()
driver.wcdmaMeas.multiEval.listPy.cdPower.edpdch.repcap_edpdChannel_set(repcap.
↪EdpdChannel.Nr1)
```

class EdpdchCls

Edpdch commands group definition. 5 total commands, 5 Subgroups, 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.wcdmaMeas.multiEval.listPy.cdPower.edpdch.clone()
```

Subgroups

6.4.1.3.2.22 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPDch<nr>:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:EDPDch<nr>
↪:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param edpdChannel

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

return

edpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.23 Current**SCPI Command :**

```
FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:EDPDch<nr>:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:EDPDch<nr>
↪:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param edpdChannel

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

return

edpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.24 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPDch<nr>:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPDch<nr>
↪:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param edpdChannel

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

return

edpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.25 Minimum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPDch<nr>:MINimum
```

class MinimumCls

Minimum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPDch<nr>
↪:MINimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param edpdChannel

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

return

edpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.26 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPDch<nr>:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(edpdChannel=EdpdChannel.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:EDPDch<nr>
↳:SDEviation
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param edpdChannel

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

return

edpdch: Comma-separated list of values, one per measured segment

6.4.1.3.2.27 Hsdpcch

class HsdpcchCls

Hsdpcch commands group definition. 5 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.2.28 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:HSDPcch:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:CDPower:HSDPcch:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.hsdpcch.average.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

hsdpcch: Comma-separated list of values, one per measured segment

6.4.1.3.2.29 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:HSDPcch:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:CDPower:HSDPcch:CURRENT
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.hsdpcch.current.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return

hsdpcch: Comma-separated list of values, one per measured segment

6.4.1.3.2.30 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:HSDPcch:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:CDPower:HSDPcch:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.hsdpcch.maximum.
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
hsdpccch: Comma-separated list of values, one per measured segment

6.4.1.3.2.31 Minimum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:HSDPcch:MINimum
```

class MinimumCls

Minimum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>  
↳:MEvaluation:LIST:CDPower:HSDPcch:MINimum  
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.hsdpcch.minimum.  
↳fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
hsdpccch: Comma-separated list of values, one per measured segment

6.4.1.3.2.32 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:HSDPcch:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>  
↳:MEvaluation:LIST:CDPower:HSDPcch:SDEViation  
value: List[float] = driver.wcdmaMeas.multiEval.listPy.cdPower.hsdpcch.  
↳standardDev.fetch()
```

Return RMS CDP and CDE vs. slot values for the HS-DPCCH for all measured list mode segments.

Suppressed linked return values: reliability

return
hsdpccch: Comma-separated list of values, one per measured segment

6.4.1.3.2.33 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: RMS CDP values for the indicated channels
- Dpdch: List[float]: RMS CDP values for the indicated channels
- Hsdpcch: List[float]: RMS CDP values for the indicated channels
- Edpcch: List[float]: RMS CDP values for the indicated channels
- Edpdch_1: List[float]: RMS CDP values for the indicated channels
- Edpdch_2: List[float]: RMS CDP values for the indicated channels
- Edpdch_3: List[float]: RMS CDP values for the indicated channels
- Edpdch_4: List[float]: RMS CDP values for the indicated channels

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:MAXimum
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.2.34 Minimum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:MINimum
```

class MinimumCls

Minimum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: RMS CDP values for the indicated channels
- Dpdch: List[float]: RMS CDP values for the indicated channels
- Hsdpcch: List[float]: RMS CDP values for the indicated channels
- Edpcch: List[float]: RMS CDP values for the indicated channels
- Edpdch_1: List[float]: RMS CDP values for the indicated channels
- Edpdch_2: List[float]: RMS CDP values for the indicated channels
- Edpdch_3: List[float]: RMS CDP values for the indicated channels
- Edpdch_4: List[float]: RMS CDP values for the indicated channels

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:MINimum
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.2.35 StandardDev**SCPI Command :**

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:CDPower:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: RMS CDP values for the indicated channels
- Dpdch: List[float]: RMS CDP values for the indicated channels

- Hsdpcch: List[float]: RMS CDP values for the indicated channels
- Edpcch: List[float]: RMS CDP values for the indicated channels
- Edpdch_1: List[float]: RMS CDP values for the indicated channels
- Edpdch_2: List[float]: RMS CDP values for the indicated channels
- Edpdch_3: List[float]: RMS CDP values for the indicated channels
- Edpdch_4: List[float]: RMS CDP values for the indicated channels

fetch() → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:CDPower:SDEviation
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.3 Modulation

class ModulationCls

Modulation commands group definition. 48 total commands, 12 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.1 Average

SCPI Command :

```
FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: Error vector magnitude RMS and peak value
- Evm_Peak: List[float]: Error vector magnitude RMS and peak value
- Mag_Error_Rms: List[float]: Magnitude error RMS value
- Mag_Error_Peak: List[float]: Magnitude error peak value
- Phase_Error_Rms: List[float]: No parameter help available
- Phase_Error_Peak: List[float]: No parameter help available
- Iq_Offset: List[float]: I/Q origin offset
- Iq_Imbalance: List[float]: I/Q imbalance
- Carrier_Freq_Err: List[float]: Carrier frequency error
- Transmit_Time_Err: List[float]: No parameter help available
- Ue_Power: List[float]: User equipment power

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:MODULATION:AVERAGE
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.3.2 Current

SCPI Command :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:MODULATION:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: Error vector magnitude RMS and peak value
- Evm_Peak: List[float]: Error vector magnitude RMS and peak value
- Mag_Error_Rms: List[float]: Magnitude error RMS value
- Mag_Error_Peak: List[float]: Magnitude error peak value

- Phase_Error_Rms: List[float]: No parameter help available
- Phase_Error_Peak: List[float]: No parameter help available
- Iq_Offset: List[float]: I/Q origin offset
- Iq_Imbalance: List[float]: I/Q imbalance
- Carrier_Freq_Err: List[float]: Carrier frequency error
- Transmit_Time_Err: List[float]: No parameter help available
- Ue_Power: List[float]: User equipment power

fetch() → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:CURRent
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.3.3 Evm

class EvmCls

Evm commands group definition. 8 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.4 Peak

class PeakCls

Peak commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.5 Average

SCPI Command :

```
FEtCh:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:EVM:PEAK:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FEtCh:WCDMa:MEASurement<instance>
↳:MEValuation:LIST:MODulation:EVM:PEAK:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.evm.peak.
↳average.fetch()
```

Return error vector magnitude peak values for all measured list mode segments.

Suppressed linked return values: reliability

return

evm_peak: Comma-separated list of values, one per measured segment

6.4.1.3.3.6 Current

SCPI Command :

```
FEtCh:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:EVM:PEAK:CURREnt
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FEtCh:WCDMa:MEASurement<instance>
↳:MEValuation:LIST:MODulation:EVM:PEAK:CURREnt
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.evm.peak.
↳current.fetch()
```

Return error vector magnitude peak values for all measured list mode segments.

Suppressed linked return values: reliability

return

evm_peak: Comma-separated list of values, one per measured segment

6.4.1.3.3.7 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:EVM:PEAK:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.evm.peak.
↪maximum.fetch()
```

Return error vector magnitude peak values for all measured list mode segments.

Suppressed linked return values: reliability

return
evm_peak: Comma-separated list of values, one per measured segment

6.4.1.3.3.8 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:EVM:PEAK:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:PEAK:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.evm.peak.
↪standardDev.fetch()
```

Return error vector magnitude peak values for all measured list mode segments.

Suppressed linked return values: reliability

return
evm_peak: Comma-separated list of values, one per measured segment

6.4.1.3.3.9 Rms

class RmsCls

Rms commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.10 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODulation:EVM:RMS:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.evm.rms.
↪average.fetch()
```

Return error vector magnitude RMS values for all measured list mode segments.

Suppressed linked return values: reliability

return

evm_rms: Comma-separated list of values, one per measured segment

6.4.1.3.3.11 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODulation:EVM:RMS:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:EVM:RMS:CURRENT
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.evm.rms.
↪current.fetch()
```

Return error vector magnitude RMS values for all measured list mode segments.

Suppressed linked return values: reliability

return
evm_rms: Comma-separated list of values, one per measured segment

6.4.1.3.3.12 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:EVM:RMS:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:EVM:RMS:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.evm.rms.
↳maximum.fetch()
```

Return error vector magnitude RMS values for all measured list mode segments.

Suppressed linked return values: reliability

return
evm_rms: Comma-separated list of values, one per measured segment

6.4.1.3.3.13 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:EVM:RMS:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:EVM:RMS:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.evm.rms.
↳standardDev.fetch()
```

Return error vector magnitude RMS values for all measured list mode segments.

Suppressed linked return values: reliability

return
evm_rms: Comma-separated list of values, one per measured segment

6.4.1.3.3.14 FreqError

class FreqErrorCls

FreqError commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.15 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODulation:FERRor:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:FERRor:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.freqError.
↳average.fetch()
```

Return carrier frequency error values for all measured list mode segments.

Suppressed linked return values: reliability

return

carrier_freq_err: Comma-separated list of values, one per measured segment

6.4.1.3.3.16 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODulation:FERRor:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:FERRor:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.freqError.
↳current.fetch()
```


Return carrier frequency error values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    carrier_freq_err: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.17 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:FERRor:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:FERRor:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.freqError.
↳maximum.fetch()
```

Return carrier frequency error values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    carrier_freq_err: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.18 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:FERRor:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:FERRor:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.freqError.
↳standardDev.fetch()
```

Return carrier frequency error values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    carrier_freq_err: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.19 IqImbalance

class IqImbalanceCls

IqImbalance commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.20 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:IQIMbalance:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.iqImbalance.
↪average.fetch()
```

Return I/Q imbalance values for all measured list mode segments.

Suppressed linked return values: reliability

return

iq_imbalance: Comma-separated list of values, one per measured segment

6.4.1.3.3.21 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:IQIMbalance:CURREnt
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:IQIMbalance:CURREnt
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.iqImbalance.
↪current.fetch()
```

Return I/Q imbalance values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    iq_imbalance: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.22 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:IQIMbalance:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:IQIMbalance:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.iqImbalance.
↳maximum.fetch()
```

Return I/Q imbalance values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    iq_imbalance: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.23 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:IQIMbalance:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:IQIMbalance:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.iqImbalance.
↳standardDev.fetch()
```

Return I/Q imbalance values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    iq_imbalance: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.24 IqOffset

class IqOffsetCls

IqOffset commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.25 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:IQOffset:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:IQOffset:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.iqOffset.
↪average.fetch()
```

Return I/Q origin offset values for all measured list mode segments.

Suppressed linked return values: reliability

return

iq_offset: Comma-separated list of values, one per measured segment

6.4.1.3.3.26 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:IQOffset:CURREnt
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:IQOffset:CURREnt
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.iqOffset.
↪current.fetch()
```

Return I/Q origin offset values for all measured list mode segments.

Suppressed linked return values: reliability

return
iq_offset: Comma-separated list of values, one per measured segment

6.4.1.3.3.27 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:IQOffset:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:IQOffset:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.iqOffset.
↳maximum.fetch()
```

Return I/Q origin offset values for all measured list mode segments.

Suppressed linked return values: reliability

return
iq_offset: Comma-separated list of values, one per measured segment

6.4.1.3.3.28 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:IQOffset:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:IQOffset:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.iqOffset.
↳standardDev.fetch()
```

Return I/Q origin offset values for all measured list mode segments.

Suppressed linked return values: reliability

return
iq_offset: Comma-separated list of values, one per measured segment

6.4.1.3.3.29 Maximum

SCPI Command :

```
FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:MODulation:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: Error vector magnitude RMS and peak value
- Evm_Peak: List[float]: Error vector magnitude RMS and peak value
- Mag_Error_Rms: List[float]: Magnitude error RMS value
- Mag_Error_Peak: List[float]: Magnitude error peak value
- Phase_Error_Rms: List[float]: No parameter help available
- Phase_Error_Peak: List[float]: No parameter help available
- Iq_Offset: List[float]: I/Q origin offset
- Iq_Imbalance: List[float]: I/Q imbalance
- Carrier_Freq_Err: List[float]: Carrier frequency error
- Transmit_Time_Err: List[float]: No parameter help available
- Ue_Power: List[float]: User equipment power

fetch() → FetchStruct

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:MODulation:MAXimum
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.3.30 Merror

class MerrorCls

Merror commands group definition. 8 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.31 Peak

class PeakCls

Peak commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.32 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:MERRor:PEAK:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:MERRor:PEAK:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.merror.peak.
↳average.fetch()
```

Return magnitude error peak values for all measured list mode segments.

Suppressed linked return values: reliability

return

mag_error_peak: Comma-separated list of values, one per measured segment

6.4.1.3.3.33 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:MERRor:PEAK:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:MODulation:MERRor:PEAK:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.merror.peak.
↪current.fetch()
```

Return magnitude error peak values for all measured list mode segments.

Suppressed linked return values: reliability

return
mag_error_peak: Comma-separated list of values, one per measured segment

6.4.1.3.3.34 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:MERRor:PEAK:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:MODulation:MERRor:PEAK:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.merror.peak.
↪maximum.fetch()
```

Return magnitude error peak values for all measured list mode segments.

Suppressed linked return values: reliability

return
mag_error_peak: Comma-separated list of values, one per measured segment

6.4.1.3.3.35 StandardDev

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODulation:MERRor:PEAK:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:MERRor:PEAK:SDEViation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.merror.peak.
↪standardDev.fetch()
```

Return magnitude error peak values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    mag_error_peak: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.36 Rms

class RmsCls

Rms commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.37 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODulation:MERRor:RMS:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.merror.rms.
↪average.fetch()
```

Return magnitude error RMS values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    mag_error_rms: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.38 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODulation:MERRor:RMS:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:MERRor:RMS:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.merror.rms.
↳current.fetch()
```

Return magnitude error RMS values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    mag_error_rms: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.39 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODulation:MERRor:RMS:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:MERRor:RMS:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.merror.rms.
↳maximum.fetch()
```

Return magnitude error RMS values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    mag_error_rms: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.40 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:MERRor:RMS:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:MERRor:RMS:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.merror.rms.
↪standardDev.fetch()
```

Return magnitude error RMS values for all measured list mode segments.

Suppressed linked return values: reliability

return

mag_error_rms: Comma-separated list of values, one per measured segment

6.4.1.3.3.41 Perror

class PerrorCls

Perror commands group definition. 8 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.42 Peak

class PeakCls

Peak commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.43 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:PERRor:PEAK:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.perror.peak.
↳average.fetch()
```

Return phase error peak values for all measured list mode segments.

Suppressed linked return values: reliability

return
phase_error_peak: Comma-separated list of values, one per measured segment

6.4.1.3.3.44 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:PERRor:PEAK:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:PERRor:PEAK:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.perror.peak.
↳current.fetch()
```

Return phase error peak values for all measured list mode segments.

Suppressed linked return values: reliability

return
phase_error_peak: Comma-separated list of values, one per measured segment

6.4.1.3.3.45 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:PERRor:PEAK:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:MODulation:PERRor:PEAK:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.perror.peak.
↪maximum.fetch()
```

Return phase error peak values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    phase_error_peak: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.46 StandardDev

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:PERRor:PEAK:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:MODulation:PERRor:PEAK:SDEViation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.perror.peak.
↪standardDev.fetch()
```

Return phase error peak values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    phase_error_peak: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.47 Rms

class RmsCls

Rms commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.48 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:PERRor:RMS:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:PERRor:RMS:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.perror.rms.
↳average.fetch()
```

Return phase error RMS values for all measured list mode segments.

Suppressed linked return values: reliability

return

phase_error_rms: Comma-separated list of values, one per measured segment

6.4.1.3.3.49 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:PERRor:RMS:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:PERRor:RMS:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.perror.rms.
↳current.fetch()
```

Return phase error RMS values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    phase_error_rms: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.50 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:PERRor:RMS:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:PERRor:RMS:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.perror.rms.
↳maximum.fetch()
```

Return phase error RMS values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    phase_error_rms: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.51 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:PERRor:RMS:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:PERRor:RMS:SDEViation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.perror.rms.
↳standardDev.fetch()
```

Return phase error RMS values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    phase_error_rms: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.52 StandardDev

SCPI Command :

```
FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:MODulation:SDEViation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: Error vector magnitude RMS and peak value
- Evm_Peak: List[float]: Error vector magnitude RMS and peak value
- Mag_Error_Rms: List[float]: Magnitude error RMS value
- Mag_Error_Peak: List[float]: Magnitude error peak value
- Phase_Error_Rms: List[float]: No parameter help available
- Phase_Error_Peak: List[float]: No parameter help available
- Iq_Offset: List[float]: I/Q origin offset
- Iq_Imbalance: List[float]: I/Q imbalance
- Carrier_Freq_Err: List[float]: Carrier frequency error
- Transmit_Time_Err: List[float]: No parameter help available
- Ue_Power: List[float]: User equipment power

fetch() → FetchStruct

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:MODulation:SDEViation
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.3.53 TtError

class TtErrorCls

TtError commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.54 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:TTError:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:TTError:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.ttError.
↪average.fetch()
```

Return transmit timing error values for all measured list mode segments.

Suppressed linked return values: reliability

return

transmit_time_err: Comma-separated list of values, one per measured segment

6.4.1.3.3.55 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:TTError:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:TTError:CURRENT
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.ttError.
↪current.fetch()
```

Return transmit timing error values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    transmit_time_err: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.56 Maximum

SCPI Command :

`FEtCh:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:TTError:MAXimum`

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FEtCh:WCDMa:MEASurement<instance>
↳:MEValuation:LIST:MODulation:TTError:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.ttError.
↳maximum.fetch()
```

Return transmit timing error values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    transmit_time_err: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.57 StandardDev

SCPI Command :

`FEtCh:WCDMa:MEASurement<instance>:MEValuation:LIST:MODulation:TTError:SDEviation`

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FEtCh:WCDMa:MEASurement<instance>
↳:MEValuation:LIST:MODulation:TTError:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.ttError.
↳standardDev.fetch()
```

Return transmit timing error values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    transmit_time_err: Comma-separated list of values, one per measured segment
```

6.4.1.3.3.58 UePower

class UePowerCls

UePower commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.3.59 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:UEPower:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:UEPower:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.uePower.
↪average.fetch()
```

Return user equipment power values for all measured list mode segments.

Suppressed linked return values: reliability

return

ue_power: Comma-separated list of values, one per measured segment

6.4.1.3.3.60 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:UEPower:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:MODulation:UEPower:CURRENT
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.uePower.
↪current.fetch()
```

Return user equipment power values for all measured list mode segments.

Suppressed linked return values: reliability

return
ue_power: Comma-separated list of values, one per measured segment

6.4.1.3.3.61 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:UEPower:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:UEPower:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.uePower.
↳maximum.fetch()
```

Return user equipment power values for all measured list mode segments.

Suppressed linked return values: reliability

return
ue_power: Comma-separated list of values, one per measured segment

6.4.1.3.3.62 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:MODulation:UEPower:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:MODulation:UEPower:SDEviation
value: List[float] = driver.wcdmaMeas.multiEval.listPy.modulation.uePower.
↳standardDev.fetch()
```

Return user equipment power values for all measured list mode segments.

Suppressed linked return values: reliability

return
ue_power: Comma-separated list of values, one per measured segment

6.4.1.3.4 Pcde

class PcdeCls

Pcde commands group definition. 8 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.4.1 Code

class CodeCls

Code commands group definition. 2 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.4.2 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:CODE:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[int]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:CODE:CURRENT
value: List[int] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

pcd_error_code_nr: Comma-separated list of values, one per measured segment

6.4.1.3.4.3 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:CODE:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[int]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:CODE:MAXimum
value: List[int] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

pcd_error_code_nr: Comma-separated list of values, one per measured segment

6.4.1.3.4.4 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: Peak code domain error.
- 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.wcdmaMeas.multiEval.listPy.pcde.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 RsCMPX_WcdmaMeas.Configure. WcdmaMeas.MultiEval.ListPy.count.

return

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

6.4.1.3.4.5 Error

class ErrorCls

Error commands group definition. 2 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.4.6 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:ERRor:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:ERRor:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.pcd.error.current.
↪ fetch()
```

Return peak code domain error values for all measured list mode segments.

Suppressed linked return values: reliability

return

pcd_error: Comma-separated list of values, one per measured segment

6.4.1.3.4.7 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:ERRor:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:ERRor:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.pcde.error.maximum.
↳ fetch()
```

Return peak code domain error values for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    pcde_error: Comma-separated list of values, one per measured segment
```

6.4.1.3.4.8 Maximum

SCPI Command :

```
FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:PCDE:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: Peak code domain error.
- 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.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure. WcdmaMeas.MultiEval.ListPy.count.

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


6.4.1.3.4.9 Phase

class PhaseCls

Phase commands group definition. 2 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.4.10 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:PHASe:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[PcdErrorPhase]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:PHASe:CURRent
value: List[enums.PcdErrorPhase] = driver.wcdmaMeas.multiEval.listPy.pcde.phase.
↳current.fetch()
```

Return the phase where the peak code domain error was measured, for all measured list mode segments.

Suppressed linked return values: reliability

return
pcd_error_phase: Comma-separated list of values, one per measured segment IPHase:
I-Signal QPHase: Q-Signal

6.4.1.3.4.11 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:PHASe:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[PcdErrorPhase]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PCDE:PHASe:MAXimum
value: List[enums.PcdErrorPhase] = driver.wcdmaMeas.multiEval.listPy.pcde.phase.
↳maximum.fetch()
```

Return the phase where the peak code domain error was measured, for all measured list mode segments.

Suppressed linked return values: reliability

```
return
    pcd_error_phase: Comma-separated list of values, one per measured segment IPHase:
    I-Signal QPHase: Q-Signal
```

6.4.1.3.5 Phd

class PhdCls

Phd commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.5.1 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PHD:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:PHD:CURRENT
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

```
return
    phd: 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 dis-
    continuity 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.
```

6.4.1.3.6 Segment<Segment>

RepCap Settings

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

class SegmentCls

Segment commands group definition. 20 total commands, 7 Subgroups, 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.wcdmaMeas.multiEval.listPy.segment.clone()
```

Subgroups

6.4.1.3.6.1 CdError

class CdErrorCls

CdError commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.6.2 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:CDERror:AVErAge
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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: 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: RMS CDE values for the indicated channels
- Dpdch: float: RMS CDE values for the indicated channels
- Hsdpcch: float: RMS CDE values for the indicated channels
- Edpcch: float: RMS CDE values for the indicated channels
- Edpdch_1: float: RMS CDE values for the indicated channels
- Edpdch_2: float: RMS CDE values for the indicated channels
- Edpdch_3: float: RMS CDE values for the indicated channels
- Edpdch_4: float: RMS CDE values for the indicated channels

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SEGMENT<nr>
↪:CDERROR:AVERAGE
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.3 Current

SCPI Command :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SEGMENT<nr>:CDERROR:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: RMS CDE values for the indicated channels
- Dpdch: float: RMS CDE values for the indicated channels
- Hsdpcch: float: RMS CDE values for the indicated channels
- Edpcch: float: RMS CDE values for the indicated channels
- Edpdch_1: float: RMS CDE values for the indicated channels
- Edpdch_2: float: RMS CDE values for the indicated channels

- Edpdch_3: float: RMS CDE values for the indicated channels
- Edpdch_4: float: RMS CDE values for the indicated channels

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↪:CDError:CURRent
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.4 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:CDError:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: RMS CDE values for the indicated channels
- Dpdch: float: RMS CDE values for the indicated channels
- Hsdpcch: float: RMS CDE values for the indicated channels
- Edpcch: float: RMS CDE values for the indicated channels
- Edpdch_1: float: RMS CDE values for the indicated channels
- Edpdch_2: float: RMS CDE values for the indicated channels
- Edpdch_3: float: RMS CDE values for the indicated channels
- Edpdch_4: float: RMS CDE values for the indicated channels

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:CDError:MAXimum
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.5 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:CDError:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: RMS CDE values for the indicated channels
- Dpdch: float: RMS CDE values for the indicated channels
- Hsdpcch: float: RMS CDE values for the indicated channels
- Edpcch: float: RMS CDE values for the indicated channels
- Edpdch_1: float: RMS CDE values for the indicated channels
- Edpdch_2: float: RMS CDE values for the indicated channels
- Edpdch_3: float: RMS CDE values for the indicated channels
- Edpdch_4: float: RMS CDE values for the indicated channels

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:CDError:SDEviation
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.6 CdPower**class CdPowerCls**

CdPower commands group definition. 5 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups**6.4.1.3.6.7 Average****SCPI Command :**

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:CDPower:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels
- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels
- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SEGMENT<nr>
↳:CDPOWER:AVERAGE
value: FetchStruct = driver.wcdmaMeas.multiEval.listPy.segment.cdPower.average.
↳fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP 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.

6.4.1.3.6.8 Current

SCPI Command :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SEGMENT<nr>:CDPOWER:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels
- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels
- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SEGMENT<nr>
↳:CDPOWER:CURRENT
value: FetchStruct = driver.wcdmaMeas.multiEval.listPy.segment.cdPower.current.
↳fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP 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.

6.4.1.3.6.9 Maximum**SCPI Command :**

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:CDPower:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels
- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels
- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↪:CDPower:MAXimum
value: FetchStruct = driver.wcdmaMeas.multiEval.listPy.segment.cdPower.maximum.
↪fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP 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.

6.4.1.3.6.10 Minimum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:CDPower:MINimum
```

class MinimumCls

Minimum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels
- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels
- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:CDPower:MINimum
value: FetchStruct = driver.wcdmaMeas.multiEval.listPy.segment.cdPower.minimum.
↳fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP 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.

6.4.1.3.6.11 StandardDev

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:CDPower:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: RMS CDP values for the indicated channels
- Dpdch: float: RMS CDP values for the indicated channels
- Hsdpcch: float: RMS CDP values for the indicated channels
- Edpcch: float: RMS CDP values for the indicated channels
- Edpdch_1: float: RMS CDP values for the indicated channels
- Edpdch_2: float: RMS CDP values for the indicated channels
- Edpdch_3: float: RMS CDP values for the indicated channels
- Edpdch_4: float: RMS CDP values for the indicated channels

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:CDPower:SDEviation
value: FetchStruct = driver.wcdmaMeas.multiEval.listPy.segment.cdPower.
↳standardDev.fetch(segment = repcap.Segment.Default)
```

Returns the RMS CDP 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.

6.4.1.3.6.12 Modulation

class ModulationCls

Modulation commands group definition. 4 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.6.13 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:MODulation:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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: 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: Error vector magnitude RMS and peak value
- Evm_Peak: float: Error vector magnitude RMS and peak value
- Mag_Error_Rms: float: Magnitude error RMS value
- Mag_Error_Peak: float: Magnitude error peak value
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- Iq_Offset: float: I/Q origin offset
- Iq_Imbalance: float: I/Q imbalance
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: User equipment power

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:MODulation:AVERage
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.14 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:MODulation:CURRENT
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: Error vector magnitude RMS and peak value
- Evm_Peak: float: Error vector magnitude RMS and peak value
- Mag_Error_Rms: float: Magnitude error RMS value
- Mag_Error_Peak: float: Magnitude error peak value
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- Iq_Offset: float: I/Q origin offset
- Iq_Imbalance: float: I/Q imbalance
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: User equipment power

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:MODulation:CURRent
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.15 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:MODulation:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: Error vector magnitude RMS and peak value
- Evm_Peak: float: Error vector magnitude RMS and peak value
- Mag_Error_Rms: float: Magnitude error RMS value
- Mag_Error_Peak: float: Magnitude error peak value
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- Iq_Offset: float: I/Q origin offset
- Iq_Imbalance: float: I/Q imbalance
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: User equipment power

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:MODulation:MAXimum
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.16 StandardDev

SCPI Command :

```
FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:MODulation:SDEviation
```

class StandardDevCls

StandardDev commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: Error vector magnitude RMS and peak value
- Evm_Peak: float: Error vector magnitude RMS and peak value
- Mag_Error_Rms: float: Magnitude error RMS value
- Mag_Error_Peak: float: Magnitude error peak value
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- Iq_Offset: float: I/Q origin offset
- Iq_Imbalance: float: I/Q imbalance
- Carrier_Freq_Err: float: No parameter help available
- Transmit_Time_Err: float: No parameter help available
- Ue_Power: float: User equipment power

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↪:MODulation:SDEviation
value: FetchStruct = driver.wcdmaMeas.multiEval.listPy.segment.modulation.
↪standardDev.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.

6.4.1.3.6.17 Pcde

class PcdeCls

Pcde commands group definition. 2 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.6.18 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:PCDE:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: Peak code domain error
- Pcd_Error_Phase: enums.PcdErrorPhase: No parameter help available
- Pcd_Error_Code_Nr: int: No parameter help available

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↪:PCDE:CURRent
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.19 Maximum

SCPI Command :

```
FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:PCDE:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: Peak code domain error
- Pcd_Error_Phase: enums.PcdErrorPhase: No parameter help available
- Pcd_Error_Code_Nr: int: No parameter help available

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↪:PCDE:MAXimum
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.20 Phd

class PhdCls

Phd commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.6.21 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:PHD:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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: 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]: 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.

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:PHD:CURRent
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.6.22 Spectrum

class SpectrumCls

Spectrum commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.6.23 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:SPECtrum:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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: 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: Power at the nominal carrier UL frequency
- Aclr_Minus_2: 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)
- Aclr_Minus_1: 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)
- Aclr_Plus_1: 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)
- Aclr_Plus_2: 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)
- Obw: float: Occupied bandwidth
- 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: User equipment power
- Emask_Margin_Had: float: No parameter help available
- Emask_Margin_Hda: float: No parameter help available

fetch(aclr_mode: AclrMode = None, segment=Segment.Default) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↳ :SPEctrum:AVERage
value: FetchStruct = driver.wcdmaMeas.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: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is 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.

6.4.1.3.6.24 Current

SCPI Command :

```
FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:SPEctrum:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: Power at the nominal carrier UL frequency
- Aclr_Minus_2: 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)
- Aclr_Minus_1: 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)
- Aclr_Plus_1: 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)

- `Aclr_Plus_2`: 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)
- `Obw`: float: Occupied bandwidth
- `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: User equipment power
- `Emask_Margin_Had`: float: No parameter help available
- `Emask_Margin_Hda`: float: No parameter help available

fetch(*aclr_mode*: *AclrMode* = *None*, *segment*=*Segment.Default*) → *FetchStruct*

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>
↪ :SPEctrum:CURRent
value: FetchStruct = driver.wcdmaMeas.multiEval.listPy.segment.spectrum.current.
↪ 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: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is 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.

6.4.1.3.6.25 Maximum

SCPI Command :

```
FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SEGment<nr>:SPEctrum:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: ‘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: 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: Power at the nominal carrier UL frequency
- Aclr_Minus_2: 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)
- Aclr_Minus_1: 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)
- Aclr_Plus_1: 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)
- Aclr_Plus_2: 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)
- Obw: float: Occupied bandwidth
- 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: User equipment power
- Emask_Margin_Had: float: No parameter help available
- Emask_Margin_Hda: float: No parameter help available

fetch(aclr_mode: AclrMode = None, segment=Segment.Default) → FetchStruct

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↪:SPECTrum:MAXimum
value: FetchStruct = driver.wcdmaMeas.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: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is 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.

6.4.1.3.6.26 UePower

class UePowerCls

UePower commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.6.27 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>:UEPower:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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: 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]: 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.

fetch(segment=Segment.Default) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SEGment<nr>
↳:UEPower:CURRent
value: FetchStruct = driver.wcdmaMeas.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.

6.4.1.3.7 Spectrum

class SpectrumCls

Spectrum commands group definition. 48 total commands, 8 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.1 Aclr

class AclrCls

Aclr commands group definition. 6 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.2 M<Minus>

RepCap Settings

```
# Range: Ch1 .. Ch2
rc = driver.wcdmaMeas.multiEval.listPy.spectrum.aclr.m.repcap_minus_get()
driver.wcdmaMeas.multiEval.listPy.spectrum.aclr.m.repcap_minus_set(repcap.Minus.Ch1)
```

class MCls

M commands group definition. 3 total commands, 3 Subgroups, 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.wcdmaMeas.multiEval.listPy.spectrum.aclr.m.clone()
```


Subgroups

6.4.1.3.7.3 Average

SCPI Command :

```
FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:M<nr>:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(aclr_mode: AclrMode = None, minus=Minus.Default) → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:M<nr>
→:AVERage
value: List[float] = driver.wcdmaMeas.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.

INTRO_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

Suppressed linked return values: reliability

param aclr_mode

ABSolute: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

param minus

optional repeated capability selector. Default value: Ch1 (settable in the interface 'M')

return

aclr: Comma-separated list of values, one per measured segment

6.4.1.3.7.4 Current

SCPI Command :

```
FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:M<nr>:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(aclr_mode: AclrMode = None, minus=Minus.Default) → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:M<nr>
→:CURRent
value: List[float] = driver.wcdmaMeas.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.

INTRO_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

Suppressed linked return values: reliability

param aclr_mode

ABSolute: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

param minus

optional repeated capability selector. Default value: Ch1 (settable in the interface 'M')

return

aclr: Comma-separated list of values, one per measured segment

6.4.1.3.7.5 Maximum**SCPI Command :**

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SPECTRUM:ACLR:M<nr>:MAXIMUM
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(aclr_mode: AclrMode = None, minus=Minus.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SPECTRUM:ACLR:M<nr>
↪:MAXIMUM
value: List[float] = driver.wcdmaMeas.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.

INTRO_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

Suppressed linked return values: reliability

param aclr_mode

ABSolute: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

param minus

optional repeated capability selector. Default value: Ch1 (settable in the interface 'M')

return

aclr: Comma-separated list of values, one per measured segment

6.4.1.3.7.6 P<Plus>

RepCap Settings

```
# Range: Ch1 .. Ch2
rc = driver.wcdmaMeas.multiEval.listPy.spectrum.aclr.p.repcap_plus_get()
driver.wcdmaMeas.multiEval.listPy.spectrum.aclr.p.repcap_plus_set(repcap.Plus.Ch1)
```

class PCl

P commands group definition. 3 total commands, 3 Subgroups, 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.wcdmaMeas.multiEval.listPy.spectrum.aclr.p.clone()
```

Subgroups

6.4.1.3.7.7 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:P<nr>:AVERage
```

class AverageCl

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(aclr_mode: AcIrMode = None, plus=Plus.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:P<nr>
↪:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.aclr.p.average.
↪fetch(aclr_mode = enums.AcIrMode.ABSolute, plus = repcap.Plus.Default)
```

Return the power of the adjacent channels for all measured list mode segments.

INTRO_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

Suppressed linked return values: reliability

param aclr_mode

ABSolute: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

param plus

optional repeated capability selector. Default value: Ch1 (settable in the interface 'P')

return

aclr: Comma-separated list of values, one per measured segment

6.4.1.3.7.8 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:P<nr>:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(aclr_mode: AclrMode = None, plus=Plus.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:P<nr>
↪:CURRent
value: List[float] = driver.wcdmaMeas.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.

INTRO_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

Suppressed linked return values: reliability

param aclr_mode

ABSolute: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

param plus

optional repeated capability selector. Default value: Ch1 (settable in the interface 'P')

return

aclr: Comma-separated list of values, one per measured segment

6.4.1.3.7.9 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:P<nr>:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(aclr_mode: AclrMode = None, plus=Plus.Default) → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:ACLR:P<nr>
↪:MAXimum
value: List[float] = driver.wcdmaMeas.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.

INTRO_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

Suppressed linked return values: reliability

param aclr_mode

ABSolute: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

param plus

optional repeated capability selector. Default value: Ch1 (settable in the interface 'P')

return

aclr: Comma-separated list of values, one per measured segment

6.4.1.3.7.10 Average**SCPI Command :**

```
FETCH:WCDMA:MEASurement<instance>:MEValuation:LIST:SPECTrum:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: Power at the nominal carrier frequency in uplink
- Aclr_Minus_2: List[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)
- Aclr_Minus_1: List[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)
- Aclr_Plus_1: List[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)
- Aclr_Plus_2: List[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)
- Obw: List[float]: Occupied bandwidth
- 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]: User equipment power
- Emask_Margin_Had: List[float]: No parameter help available
- Emask_Margin_Hda: List[float]: No parameter help available

fetch(aclr_mode: AclrMode = None) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:AVERage
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.ListPy.count.

param aclr_mode

ABSolute: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

return

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

6.4.1.3.7.11 Cpower

class CpowerCls

Cpower commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.12 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:CPower:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:CPOWer:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.cpower.average.
↳fetch()
```

Return the power at the nominal carrier frequency for all measured list mode segments.

Suppressed linked return values: reliability

return

carrier_power: Comma-separated list of values, one per measured segment

6.4.1.3.7.13 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:CPOWer:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:CPOWer:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.cpower.current.
↳fetch()
```

Return the power at the nominal carrier frequency for all measured list mode segments.

Suppressed linked return values: reliability

return

carrier_power: Comma-separated list of values, one per measured segment

6.4.1.3.7.14 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:CPOWer:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:CPOWer:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.cpower.maximum.
↳fetch()
```

Return the power at the nominal carrier frequency for all measured list mode segments.

Suppressed linked return values: reliability

return

carrier_power: Comma-separated list of values, one per measured segment

6.4.1.3.7.15 Current

SCPI Command :

`FETCh:WCDMa:MEASurement<instance>:MEValuation:LIST:SPECTrum:CURRent`

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: Power at the nominal carrier frequency in uplink
- Aclr_Minus_2: List[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)
- Aclr_Minus_1: List[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)
- Aclr_Plus_1: List[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)
- Aclr_Plus_2: List[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)
- Obw: List[float]: Occupied bandwidth
- 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]: User equipment power
- Emask_Margin_Had: List[float]: No parameter help available
- Emask_Margin_Hda: List[float]: No parameter help available

fetch(aclr_mode: AclrMode = None) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SPECTRUM:CURRENT
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.ListPy.count.

param aclr_mode

ABSolute: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

return

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

6.4.1.3.7.16 Emask

class EmaskCls

Emask commands group definition. 30 total commands, 10 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.17 Ab

class AbCls

Ab commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.18 Average

SCPI Command :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SPECTRUM:EMASK:AB:AVERAGE
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:AB:AVERAge
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.19 Current**SCPI Command :**

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:AB:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:AB:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.20 Maximum**SCPI Command :**

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:AB:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:AB:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.21 Ba

class BaCls

Ba commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.22 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:BA:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:BA:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.23 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:SPECTrum:EMASk:BA:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳ :MEValuation:LIST:SPECTrum:EMASk:BA:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.24 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:SPECTrum:EMASk:BA:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳ :MEValuation:LIST:SPECTrum:EMASk:BA:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.25 Bc

class BcCls

Bc commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.26 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:BC:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:BC:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.27 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:BC:CURREnt
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:BC:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.28 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:BC:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:BC:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.29 Cb

class CbCls

Cb commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.30 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:CB:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:CB:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.31 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:CB:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:CB:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return
emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.32 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:CB:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>  
↳:MEvaluation:LIST:SPECTrum:EMASk:CB:MAXimum  
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return
emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.33 Cd

class CdCls

Cd commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.34 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:CD:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:CD:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.35 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:CD:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:CD:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.36 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:CD:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASK:CD:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.37 Dc

class DcCls

Dc commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.38 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASK:DC:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASK:DC:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.39 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:SPECTrum:EMASk:DC:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEValuation:LIST:SPECTrum:EMASk:DC:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.40 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:SPECTrum:EMASk:DC:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEValuation:LIST:SPECTrum:EMASk:DC:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.41 Ef

class EfCls

Ef commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.42 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:EF:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:EF:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.43 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:EF:CURREnt
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>
↳:MEVALUATION:LIST:SPECTRUM:EMASK:EF:CURRENT
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.44 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASUREMENT<instance>:MEVALUATION:LIST:SPECTRUM:EMASK:EF:MAXIMUM
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>
↳:MEVALUATION:LIST:SPECTRUM:EMASK:EF:MAXIMUM
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.45 Fe

class FeCls

Fe commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.46 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:FE:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:FE:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.47 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:FE:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:FE:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return
emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.48 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:FE:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:LIST:SPECTrum:EMASk:FE:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return
emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.49 Had

class HadCls

Had commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.50 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:HAD:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASK:HAD:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.51 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASK:HAD:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASK:HAD:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.52 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASK:HAD:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASK:HAD:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.53 Hda

class HdaCls

Hda commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.54 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:EMASk:HDA:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:EMASk:HDA:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.55 Current

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:SPECTrum:EMASk:HDA:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:SPECTrum:EMASk:HDA:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.56 Maximum

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:LIST:SPECTrum:EMASk:HDA:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEValuation:LIST:SPECTrum:EMASk:HDA:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_margin: Comma-separated list of values, one per measured segment

6.4.1.3.7.57 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class FetchStruct

Response structure. Fields:

- Reliability: int: '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]: 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]: Power at the nominal carrier frequency in uplink
- Aclr_Minus_2: List[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)
- Aclr_Minus_1: List[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)
- Aclr_Plus_1: List[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)
- Aclr_Plus_2: List[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)
- Obw: List[float]: Occupied bandwidth
- 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]: User equipment power
- Emask_Margin_Had: List[float]: No parameter help available
- Emask_Margin_Hda: List[float]: No parameter help available

fetch(aclr_mode: AclrMode = None) → FetchStruct

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:MAXimum
value: FetchStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.ListPy.count.

param aclr_mode

ABSolute: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

return

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

6.4.1.3.7.58 Obw

class ObwCls

Obw commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.59 Average

SCPI Command :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:OBW:AVERAge
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:LIST:SPECTrum:OBW:AVERAge
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.obw.average.
↪ fetch()
```

Return the occupied bandwidth for all measured list mode segments.

Suppressed linked return values: reliability

return

obw: Comma-separated list of values, one per measured segment

6.4.1.3.7.60 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:OBW:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:OBW:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.obw.current.
↪ fetch()
```

Return the occupied bandwidth for all measured list mode segments.

Suppressed linked return values: reliability

return

obw: Comma-separated list of values, one per measured segment

6.4.1.3.7.61 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:OBW:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:OBW:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.obw.maximum.
↪ fetch()
```

Return the occupied bandwidth for all measured list mode segments.

Suppressed linked return values: reliability

return

obw: Comma-separated list of values, one per measured segment

6.4.1.3.7.62 UePower

class UePowerCls

UePower commands group definition. 3 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.7.63 Average

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:UEPower:AVERage
```

class AverageCls

Average commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:UEPower:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.uePower.average.
↪fetch()
```

Return the UE power for all measured list mode segments.

Suppressed linked return values: reliability

return

ue_power: Comma-separated list of values, one per measured segment

6.4.1.3.7.64 Current

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:UEPower:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:LIST:SPECTrum:UEPower:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.uePower.current.
↪fetch()
```

Return the UE power for all measured list mode segments.

Suppressed linked return values: reliability

return

ue_power: Comma-separated list of values, one per measured segment

6.4.1.3.7.65 Maximum

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SPECTrum:UEPower:MAXimum
```

class MaximumCls

Maximum commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
→:MEvaluation:LIST:SPECTrum:UEPower:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.listPy.spectrum.uePower.maximum.
→fetch()
```

Return the UE power for all measured list mode segments.

Suppressed linked return values: reliability

return
ue_power: Comma-separated list of values, one per measured segment

6.4.1.3.8 Sreliability

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SRELiability
```

class SreliabilityCls

Sreliability commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[int]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:LIST:SRELiability
value: List[int] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return
seg_reliability: 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.

6.4.1.3.9 UePower

class UePowerCls

UePower commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.3.9.1 Current

SCPI Command :

```
FEtCh:WCDMa:MEASurement<instance>:MEValuation:LIST:UEPower:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch() → List[float]

```
# SCPI: FEtCh:WCDMa:MEASurement<instance>:MEValuation:LIST:UEPower:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.listPy.uePower.current.fetch()
```

Returns the UE power vs. slot results in list mode.

Suppressed linked return values: reliability

return

ue_power: 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.

6.4.1.4 Modulation

class ModulationCls

Modulation commands group definition. 6 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.4.1 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 PhDhsDpcchCls

PhDhsDpcch commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Overall_Max_Ph_D: float or bool: No parameter help available
- Measure_Points: float or bool: No parameter help available
- Count_Dyn_Limit: float or bool: Number of results exceeding the limit
- Ratio_Dyn_Limit: float or bool: Percentage of results exceeding the limit
- Meas_Point_Acurr: float or bool: No parameter help available
- Meas_Point_Amax: float or bool: No parameter help available
- Meas_Point_Bcurr: float or bool: No parameter help available
- Meas_Point_Bmax: float or bool: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Overall_Max_Ph_D: float: No parameter help available
- Measure_Points: int: No parameter help available
- Count_Dyn_Limit: int: Number of results exceeding the limit
- Ratio_Dyn_Limit: float: Percentage of results exceeding the limit
- 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.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.PhsDpcch.set) . 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.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.PhsDpcch.set) . 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.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.PhsDpcch.set) . 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.

6.4.1.4.2 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 UephdCls

Uephd commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Overall_Max_Ph_D: float or bool: Overall maximum phase discontinuity
- Overall_Min_Dist: float or bool: The overall minimum slot distance between the two results exceeding the dynamic limit
- Count_Upper_Limit: float or bool: Number of results exceeding the upper limit
- Count_Dyn_Limit: float or bool: The number of results exceeding the dynamic limit

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Overall_Max_Ph_D: float: Overall maximum phase discontinuity
- Overall_Min_Dist: int: The overall minimum slot distance between the two results exceeding the dynamic limit
- Count_Upper_Limit: int: Number of results exceeding the upper limit
- Count_Dyn_Limit: int: The number of results exceeding the dynamic limit

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEvaluation:MODulation:UEPHd
value: CalculateStruct = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Phd.set. 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.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Phd.set. 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.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Phd.set. 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.

6.4.1.5 Pcdde

class PcddeCls

Pcdde commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.5.1 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:PCDE:CURRent
FETCH:WCDMa:MEASurement<instance>:MEvaluation:PCDE:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Pcd_Error: float: Peak code domain error
- Pcd_Error_Phase: enums.PcdErrorPhase: Phase where the peak code domain error was measured. IPHase: I-Signal QPHase: Q-Signal
- Pcd_Error_Code_Nr: int: Code number for which the PCDE was measured.

fetch() → ResultData

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:PCDE:CURRent
value: ResultData = driver.wcdmaMeas.multiEval.pcdde.current.fetch()
```

Returns the peak code domain error (PCDE) results. In addition to the current PCDE value, the maximum PCDE value can be retrieved.

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.wcdmaMeas.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.

return

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

6.4.1.5.2 Maximum

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:PCDE:MAXimum
FETCh:WCDMA:MEASurement<instance>:MEvaluation:PCDE:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Pcd_Error: float: Peak code domain error
- Pcd_Error_Phase: enums.PcdErrorPhase: Phase where the peak code domain error was measured.
IPHase: I-Signal QPHase: Q-Signal
- Pcd_Error_Code_Nr: int: Code number for which the PCDE was measured.

fetch() → ResultData

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:PCDE:MAXimum
value: ResultData = driver.wcdmaMeas.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.

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.wcdmaMeas.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.

return

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

6.4.1.6 Spectrum

class SpectrumCls

Spectrum commands group definition. 9 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.6.1 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 AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Carrier_Power: float or bool: Power at the nominal carrier UL frequency
- Aclr_Minus_2: float or bool: 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)
- Aclr_Minus_1: float or bool: 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)
- Aclr_Plus_1: float or bool: 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)
- Aclr_Plus_2: float or bool: 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)
- Obw: float or bool: Occupied bandwidth
- Sem_Margin_Abij: float or bool: No parameter help available
- Sem_Margin_Bcjk: float or bool: No parameter help available
- Sem_Margin_Cdkl: float or bool: No parameter help available
- Sem_Margin_Efmn: float or bool: No parameter help available
- Sem_Margin_Fenm: float or bool: No parameter help available
- Sem_Margin_Dclk: float or bool: No parameter help available
- Sem_Margin_Cbkj: float or bool: No parameter help available

- Sem_Margin_Baji: float or bool: No parameter help available
- Ue_Power: enums.ResultStatus2: User equipment power
- Emask_Margin_Ad: float or bool: No parameter help available
- Emask_Margin_Da: float or bool: No parameter help available
- Carrier_Power_L: enums.ResultStatus2: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection
- Carrier_Power_R: enums.ResultStatus2: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Carrier_Power: float: Power at the nominal carrier UL frequency
- Aclr_Minus_2: 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)
- Aclr_Minus_1: 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)
- Aclr_Plus_1: 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)
- Aclr_Plus_2: 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)
- Obw: float: Occupied bandwidth
- 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: User equipment power
- 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: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection
- Carrier_Power_R: float: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEvaluation:SPECTrum:AVERage
value: CalculateStruct = driver.wcdmaMeas.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. 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: AclrMode = None) → ResultData

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:SPECTrum:AVERage
value: ResultData = driver.wcdmaMeas.multiEval.spectrum.average.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. 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: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

return

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

read(aclr_mode: AclrMode = None) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:SPECTrum:AVERage
value: ResultData = driver.wcdmaMeas.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. 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: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

return

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

6.4.1.6.2 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 CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Carrier_Power: float or bool: Power at the nominal carrier UL frequency
- Aclr_Minus_2: float or bool: 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)
- Aclr_Minus_1: float or bool: 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)
- Aclr_Plus_1: float or bool: 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)
- Aclr_Plus_2: float or bool: 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)
- Obw: float or bool: Occupied bandwidth
- Sem_Margin_Abij: float or bool: No parameter help available
- Sem_Margin_Bcjk: float or bool: No parameter help available
- Sem_Margin_Cdkl: float or bool: No parameter help available
- Sem_Margin_Efmn: float or bool: No parameter help available
- Sem_Margin_Fenm: float or bool: No parameter help available
- Sem_Margin_Delk: float or bool: No parameter help available
- Sem_Margin_Cbkj: float or bool: No parameter help available
- Sem_Margin_Baji: float or bool: No parameter help available
- Ue_Power: enums.ResultStatus2: User equipment power
- Emask_Margin_Ad: float or bool: No parameter help available
- Emask_Margin_Da: float or bool: No parameter help available
- Carrier_Power_L: enums.ResultStatus2: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection

- Carrier_Power_R: enums.ResultStatus2: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Carrier_Power: float: Power at the nominal carrier UL frequency
- Aclr_Minus_2: 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)
- Aclr_Minus_1: 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)
- Aclr_Plus_1: 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)
- Aclr_Plus_2: 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)
- Obw: float: Occupied bandwidth
- 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: User equipment power
- 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: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection

- **Carrier_Power_R**: float: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:MEValuation:SPECTrum:CURRent
value: CalculateStruct = driver.wcdmaMeas.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. 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: AclrMode = None) → ResultData

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:SPECTrum:CURRent
value: ResultData = driver.wcdmaMeas.multiEval.spectrum.current.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. 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: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

return

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

read(aclr_mode: AclrMode = None) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:SPECTrum:CURRent
value: ResultData = driver.wcdmaMeas.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. 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: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

return

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

6.4.1.6.3 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 MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Carrier_Power: float or bool: Power at the nominal carrier UL frequency
- Aclr_Minus_2: float or bool: 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)
- Aclr_Minus_1: float or bool: 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)
- Aclr_Plus_1: float or bool: 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)
- Aclr_Plus_2: float or bool: 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)
- Obw: float or bool: Occupied bandwidth
- Sem_Margin_Abij: float or bool: No parameter help available
- Sem_Margin_Bcjk: float or bool: No parameter help available
- Sem_Margin_Cdkl: float or bool: No parameter help available
- Sem_Margin_Efmn: float or bool: No parameter help available
- Sem_Margin_Fenm: float or bool: No parameter help available
- Sem_Margin_Dclk: float or bool: No parameter help available
- Sem_Margin_Cbkj: float or bool: No parameter help available
- Sem_Margin_Baji: float or bool: No parameter help available
- Ue_Power: enums.ResultStatus2: User equipment power
- Emask_Margin_Ad: float or bool: No parameter help available
- Emask_Margin_Da: float or bool: No parameter help available
- Carrier_Power_L: enums.ResultStatus2: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection
- Carrier_Power_R: enums.ResultStatus2: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'

- Carrier_Power: float: Power at the nominal carrier UL frequency
- Aclr_Minus_2: 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)
- Aclr_Minus_1: 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)
- Aclr_Plus_1: 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)
- Aclr_Plus_2: 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)
- Obw: float: Occupied bandwidth
- 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: User equipment power
- 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: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection
- Carrier_Power_R: float: Power at the nominal carrier frequency; left/right carrier of the dual carrier HSPA connection

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:MEvaluation:SPECTrum:MAXimum
value: CalculateStruct = driver.wcdmaMeas.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. 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*: *AcIrMode* = *None*) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:SPECTrum:MAXimum
value: ResultData = driver.wcdmaMeas.multiEval.spectrum.maximum.fetch(aclr_mode_
↳= enums.AcIrMode.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. 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: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

return

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

read(*aclr_mode*: *AcIrMode* = *None*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:SPECTrum:MAXimum
value: ResultData = driver.wcdmaMeas.multiEval.spectrum.maximum.read(aclr_mode_
↳= enums.AcIrMode.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. 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: ACLR power is displayed in dBm as an absolute value. RELative: ACLR power is displayed in dB relative to carrier power.

return

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

6.4.1.7 State

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:STATE
```

class StateCls

State commands group definition. 2 total commands, 1 Subgroups, 1 group commands

fetch(*timeout: float = None, target_main_state: TargetMainState = None, target_sync_state: TargetSyncState = None*) → ResourceState

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:STATE
value: enums.ResourceState = driver.wcdmaMeas.multiEval.state.fetch(timeout = 1.
↪0, target_main_state = enums.TargetMainState.OFF, target_sync_state = enums.
↪TargetSyncState.ADJusted)
```

Queries the main measurement state. Without query parameters, the state is returned immediately. With query parameters, the state is returned when the <TargetMainState> and the <TargetSyncState> are reached or when the <Timeout> expires.

param timeout

No help available

param target_main_state

Target MainState for the query Default is RUN.

param target_sync_state

Target SyncState for the query Default is ADJ.

return

state: Current state or target state of ongoing state transition OFF: measurement off
RUN: measurement running RDY: measurement completed

Cloning the Group

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

Subgroups

6.4.1.7.1 All

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:STATE:ALL
```

class AllCls

All commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(*timeout: float = None, target_main_state: TargetMainState = None, target_sync_state: TargetSyncState = None*) → List[ResourceState]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:MEvaluation:STATE:ALL
value: List[enums.ResourceState] = driver.wcdmaMeas.multiEval.state.all.
↪ fetch(timeout = 1.0, target_main_state = enums.TargetMainState.OFF, target_
↪ sync_state = enums.TargetSyncState.ADJusted)
```

Queries the main measurement state and the measurement substates. Without query parameters, the states are returned immediately. With query parameters, the states are returned when the <TargetMainState> and the <TargetSyncState> are reached or when the <Timeout> expires.

param timeout

No help available

param target_main_state

Target MainState for the query Default is RUN.

param target_sync_state

Target SyncState for the query Default is ADJ.

return

state: No help available

6.4.1.8 Trace

class TraceCls

Trace commands group definition. 61 total commands, 9 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.1 Carrier<Carrier>

RepCap Settings

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

class CarrierCls

Carrier commands group definition. 1 total commands, 1 Subgroups, 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.wcdmaMeas.multiEval.trace.carrier.clone()
```

Subgroups

6.4.1.8.1.1 Perror

class PerrorCls

Perror commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.1.2 Rms

class RmsCls

Rms commands group definition. 1 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.multiEval.trace.carrier.perror.rms.clone()
```

Subgroups

6.4.1.8.1.3 Current

SCPI Command :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:CARRier<carrier>:PERRor[:RMS]:CURRent
```

class CurrentCls

Current commands group definition. 1 total commands, 0 Subgroups, 1 group commands

read(carrier=Carrier.Default) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:CARRier<carrier>
↪:PERRor[:RMS]:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Mperiod. modulation) . The number of results depends on the measurement length (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

phase_error: RMS phase error trace results, one result per measured slot or half-slot

6.4.1.8.2 CdeMonitor

class CdeMonitorCls

CdeMonitor commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.2.1 Isignal

class IsignalCls

Isignal commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.2.2 Current

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACe:CDEMonitor:ISIGnal:CURRent
READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:CDEMonitor:ISIGnal:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪:MEvaluation:TRACe:CDEMonitor:ISIGnal:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

isignal: One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) .

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>
↪:MEvaluation:TRACe:CDEMonitor:ISIGnal:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

isignal: One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) .

6.4.1.8.2.3 Qsignal**class QsignalCls**

Qsignal commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.2.4 Current

SCPI Commands :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:CDEMonitor:QSIGnal:CURRent
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:CDEMonitor:QSIGnal:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
→:MEvaluation:TRACe:CDEMonitor:QSIGnal:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

qsignal: One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) .

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>
→:MEvaluation:TRACe:CDEMonitor:QSIGnal:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

qsignal: One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) .

6.4.1.8.3 CdpMonitor

class CdpMonitorCls

CdpMonitor commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.3.1 Isignal

class IsignalCls

Isignal commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.3.2 Current

SCPI Commands :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:CDPMonitor:ISIGnal:CURRent
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:CDPMonitor:ISIGnal:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:TRACe:CDPMonitor:ISIGnal:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

isignal: One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) .

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>
↳:MEvaluation:TRACe:CDPMonitor:ISIGnal:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

isignal: One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) .

6.4.1.8.3.3 Qsignal

class QsignalCls

Qsignal commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.3.4 Current

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACe:CDPMonitor:QSIGnal:CURRent
READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:CDPMonitor:QSIGnal:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:CDPMonitor:QSIGnal:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

qsignal: One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) .

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:CDPMonitor:QSIGnal:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

qsignal: One value per code channel. The number of values/channels corresponds to the spreading factor (e.g. 8 values/channels for SF8) .

6.4.1.8.4 Emask

class EmaskCls

Emask commands group definition. 30 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.4.1 HkfLeft

class HkfLeftCls

HkfLeft commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.4.2 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:HKFLeft:AVERage
FETCh:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:HKFLeft:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↳:MEvaluation:TRACe:EMASk:HKFLeft:AVERAge
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kleft: 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)

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:HKFLeft:AVERAge
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kleft: 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)

6.4.1.8.4.3 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:HKFLeft:CURRent
FETCh:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:HKFLeft:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]


```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↳:MEvaluation:TRACe:EMASk:HKFLeft:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kleft: 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)

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:HKFLeft:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kleft: 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)

6.4.1.8.4.4 Maximum

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:HKFLeft:MAXimum
FETCh:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:HKFLeft:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↳:MEvaluation:TRACe:EMASk:HKFLeft:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kleft: 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)

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:HKFLeft:MAXimum
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kleft: 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)

6.4.1.8.4.5 HkfRight

class HkfRightCls

HkfRight commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.4.6 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:HKFRight:AVERage
FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:HKFRight:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>
↳:MEvaluation:TRACe:EMASk:HKFRight:AVERage
value: List[float] = driver.wcdmaMeas.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 HKFLLeft and HKFRRight) . 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kright: 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)

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↳:MEvaluation:TRACe:EMASk:HKFRight:AVERage
value: List[float] = driver.wcdmaMeas.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 HKFLLeft and HKFRRight) . 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kright: 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)

6.4.1.8.4.7 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:HKFRight:CURRent
FETCh:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:HKFRight:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↳:MEValuation:TRACe:EMASk:HKFRight:CURRent
value: List[float] = driver.wcdmaMeas.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 HKFLLeft and HKFRRight) . 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kright: 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)

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↳:MEValuation:TRACe:EMASk:HKFRight:CURRent
value: List[float] = driver.wcdmaMeas.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 HKFLLeft and HKFRRight) . 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kright: 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)

6.4.1.8.4.8 Maximum

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:HKFRRight:MAXimum
FETCh:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:HKFRRight:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↳:MEValuation:TRACe:EMASk:HKFRRight:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.trace.emask.hkfrRight.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 HKFLLeft and HKFRRight) . 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kright: 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)

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>
↳:MEValuation:TRACe:EMASk:HKFRRight:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.trace.emask.hkfrRight.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 HKFLLeft and HKFRRight) . 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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.Limit.Emask.Absolute.set) .

Suppressed linked return values: reliability

return

emask_100_kright: 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)

6.4.1.8.4.9 Kfilter

class KfilterCls

Kfilter commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.4.10 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:KFILter:AVERAge
FETCh:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:KFILter:AVERAge
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:KFILter:AVERAge
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_30_k: 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 $f = (fC2 - fC1) / 2$.

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:MEValuation:TRACe:EMASk:KFILter:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_30_k: 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 $f = (fC2 - fC1) / 2$.

6.4.1.8.4.11 Current

SCPI Commands :

```
READ:WCDma:MEASurement<instance>:MEValuation:TRACe:EMASk:KFILter:CURRent
FETCh:WCDma:MEASurement<instance>:MEValuation:TRACe:EMASk:KFILter:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:KFILter:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_30_k: 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 $f = (fC2 - fC1) / 2$.

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:KFILter:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_30_k: 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 $f = (fC2 - fC1) / 2$.

6.4.1.8.4.12 Maximum

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:KFILter:MAXimum
FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:KFILter:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↪ :MEvaluation:TRACe:EMASk:KFILter:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_30_k: 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 $f = (fC2 - fC1) / 2$.

read() → List[float]


```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:KFILter:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_30_k: 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 $f = (fC2 - fC1) / 2$.

6.4.1.8.4.13 MfLeft

class MfLeftCls

MfLeft commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.4.14 Average

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLet:AVERage
FETCh:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLet:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLet:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mleft: 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

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEValuation:TRACe:EMASk:MFLeft:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mleft: 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

6.4.1.8.4.15 Current

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEValuation:TRACe:EMASk:MFLeft:CURRent
FETCh:WCDMA:MEASurement<instance>:MEValuation:TRACe:EMASk:MFLeft:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:MEValuation:TRACe:EMASk:MFLeft:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mleft: 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

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLeft:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mleft: 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

6.4.1.8.4.16 Maximum

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLeft:MAXimum
FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLeft:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLeft:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mleft: 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

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFLeft:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mleft: 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

6.4.1.8.4.17 MfRight**class MfRightCls**

MfRight commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.4.18 Average

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFRight:AVERage
FETCh:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFRight:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEvaluation:TRACe:EMASk:MFRight:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mrigh: 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

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:EMASk:MFRight:AVERage
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mright: 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

6.4.1.8.4.19 Current**SCPI Commands :**

```
READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:MFRight:CURRent
FETCh:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:MFRight:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:MFRight:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mright: 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

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASk:MFRight:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mright: 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

6.4.1.8.4.20 Maximum

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEValuation:TRACe:EMASk:MFRight:MAXimum
FETCh:WCDMA:MEASurement<instance>:MEValuation:TRACe:EMASk:MFRight:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>
↪:MEValuation:TRACe:EMASk:MFRight:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mright: 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

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEValuation:TRACe:EMASk:MFRight:MAXimum
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

emask_1_mright: 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

6.4.1.8.5 EvMagnitude

class EvMagnitudeCls

EvMagnitude commands group definition. 6 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.5.1 Chip

class ChipCls

Chip commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.5.2 Average

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEValuation:TRACe:EVMagnitude:CHIP:AVERage
FETCh:WCDMA:MEASurement<instance>:MEValuation:TRACe:EVMagnitude:CHIP:AVERage
```


class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:TRACe:EVMagnitude:CHIP:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.trace.evMagnitude.chip.average.
↳fetch()
```

Returns the values of the RMS EVM vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return

evm_chip: No help available

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>
↳:MEvaluation:TRACe:EVMagnitude:CHIP:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.trace.evMagnitude.chip.average.
↳read()
```

Returns the values of the RMS EVM vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return

evm_chip: No help available

6.4.1.8.5.3 Current**SCPI Commands :**

```
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EVMagnitude:CHIP:CURRent
FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EVMagnitude:CHIP:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>
↳:MEvaluation:TRACe:EVMagnitude:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.trace.evMagnitude.chip.current.
↳fetch()
```

Returns the values of the RMS EVM vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return
evm_chip: No help available

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>
↳:MEvaluation:TRACe:EVMagnitude:CHIP:CURRENT
value: List[float] = driver.wcdmaMeas.multiEval.trace.evMagnitude.chip.current.
↳read()
```

Returns the values of the RMS EVM vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return
evm_chip: No help available

6.4.1.8.5.4 Maximum

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EVMagnitude:CHIP:MAXimum
FETCh:WCDMA:MEASurement<instance>:MEvaluation:TRACe:EVMagnitude:CHIP:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMA:MEASurement<instance>
↳:MEvaluation:TRACe:EVMagnitude:CHIP:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.trace.evMagnitude.chip.maximum.
↳fetch()
```

Returns the values of the RMS EVM vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return
evm_chip: No help available

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>
↳:MEvaluation:TRACe:EVMagnitude:CHIP:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.trace.evMagnitude.chip.maximum.
↳read()
```

Returns the values of the RMS EVM vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas.Configure. WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

```
return
    evm_chip: No help available
```

6.4.1.8.6 Iq

class IqCls

Iq commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.6.1 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:IQ:CURRent
FETCh:WCDMa:MEASurement<instance>:MEValuation:TRACe:IQ:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: '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.wcdmaMeas.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.wcdmaMeas.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.

6.4.1.8.7 Merror

class MerrorCls

Merror commands group definition. 6 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.7.1 Chip

class ChipCls

Chip commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.7.2 Average

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACe:MERRor:CHIP:AVERage
READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:MERRor:CHIP:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACe:MERRor:CHIP:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.trace.merror.chip.average.
    ↪ fetch()
```

Returns the values of the magnitude error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return
mag_error_chip: No help available

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.trace.merror.chip.average.read()
```

Returns the values of the magnitude error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return
mag_error_chip: No help available

6.4.1.8.7.3 Current

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:CURRent
READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.trace.merror.chip.current.
    ↪ fetch()
```

Returns the values of the magnitude error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return
mag_error_chip: No help available

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.trace.merror.chip.current.read()
```

Returns the values of the magnitude error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return
mag_error_chip: No help available

6.4.1.8.7.4 Maximum

SCPI Commands :

```
FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:MERRor:CHIP:MAXimum
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:MERRor:CHIP:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:MERRor:CHIP:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.trace.merror.chip.maximum.
    ↪ fetch()
```

Returns the values of the magnitude error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

```
return
    mag_error_chip: No help available
```

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:MERRor:CHIP:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.trace.merror.chip.maximum.read()
```

Returns the values of the magnitude error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

```
return
    mag_error_chip: No help available
```

6.4.1.8.8 Perror

class PerrorCls

Perror commands group definition. 6 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.8.1 Chip

class ChipCls

Chip commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.8.2 Average

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:AVERage
READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.trace.perror.chip.average.
↳ fetch()
```

Returns the values of the RMS phase error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return
phase_error_chip: No help available

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:PERRor:CHIP:AVERage
value: List[float] = driver.wcdmaMeas.multiEval.trace.perror.chip.average.read()
```

Returns the values of the RMS phase error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

return
phase_error_chip: No help available

6.4.1.8.8.3 Current

SCPI Commands :

```

FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:PERRor:CHIP:CURRent
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:PERRor:CHIP:CURRent

```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```

# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:PERRor:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.trace.perror.chip.current.
↳ fetch()

```

Returns the values of the RMS phase error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

```

return
    phase_error_chip: No help available

```

read() → List[float]

```

# SCPI: READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:PERRor:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.multiEval.trace.perror.chip.current.read()

```

Returns the values of the RMS phase error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

```

return
    phase_error_chip: No help available

```

6.4.1.8.8.4 Maximum

SCPI Commands :

```

FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:PERRor:CHIP:MAXimum
READ:WCDMA:MEASurement<instance>:MEvaluation:TRACe:PERRor:CHIP:MAXimum

```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```

# SCPI: FETCH:WCDMA:MEASurement<instance>:MEvaluation:TRACe:PERRor:CHIP:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.trace.perror.chip.maximum.
↳ fetch()

```


Returns the values of the RMS phase error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

```
return
    phase_error_chip: No help available
```

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:PERror:CHIP:MAXimum
value: List[float] = driver.wcdmaMeas.multiEval.trace.perror.chip.maximum.read()
```

Returns the values of the RMS phase error vs chip traces, measured in the preselected slot (see method RsCMPX_WcdmaMeas. Configure.WcdmaMeas.MultiEval.pslot) . One value per chip is returned. The results of the current, average and maximum traces can be retrieved.

Suppressed linked return values: reliability

```
return
    phase_error_chip: No help available
```

6.4.1.8.9 Phd

class PhdCls

Phd commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.1.8.9.1 Current

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACe:PHD:CURRent
READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:PHD:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACe:PHD:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount) .

INTRO_CMD_HELP: The meaning of the value depends on the measurement period (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

return
phase_disc: One value per measured slot

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:MEvaluation:TRACe:PHD:CURRent
value: List[float] = driver.wcdmaMeas.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 RsCMPX_WcdmaMeas.Configure.WcdmaMeas.MultiEval.msCount) .

INTRO_CMD_HELP: The meaning of the value depends on the measurement period (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.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.

Suppressed linked return values: reliability

return
phase_disc: One value per measured slot

6.4.2 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 OlpControlCls

OlpControl commands group definition. 10 total commands, 2 Subgroups, 6 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: No parameter help available

- Olpc_1: float or bool: No parameter help available
- Olpc_2: float or bool: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: No parameter help available
- Ue_Pwr_C_1: float: No parameter help available
- Olpc_1: float: No parameter help available
- Slot_No_C_1: int: No parameter help available
- Olpc_2: float: No parameter help available
- Slot_No_C_2: int: No parameter help available

abort(opc_timeout_ms: int = -1) → None

```
# SCPI: ABORt:WCDMa:MEASurement<instance>:OLPControl
driver.wcdmaMeas.olpControl.abort()
```

No command help available

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:OLPControl
value: CalculateStruct = driver.wcdmaMeas.olpControl.calculate()
```

No command help available

return

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

fetch() → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:OLPControl
value: ResultData = driver.wcdmaMeas.olpControl.fetch()
```

No command help available

return

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

initiate(opc_timeout_ms: int = -1) → None

```
# SCPI: INITiate:WCDMa:MEASurement<instance>:OLPControl
driver.wcdmaMeas.olpControl.initiate()
```

No command help available

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

read() → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:OLPControl
value: ResultData = driver.wcdmaMeas.olpControl.read()
```

No command help available

return

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

stop(opc_timeout_ms: int = -1) → None

```
# SCPI: STOP:WCDMa:MEASurement<instance>:OLPControl
driver.wcdmaMeas.olpControl.stop()
```

No command help available

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

Cloning the Group

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

Subgroups

6.4.2.1 Carrier<CARRierExt>

RepCap Settings

```
# Range: Nr1 .. Nr32
rc = driver.wcdmaMeas.olpControl.carrier.repcap_cARRierExt_get()
driver.wcdmaMeas.olpControl.carrier.repcap_cARRierExt_set(repcap.CARRierExt.Nr1)
```

class CarrierCls

Carrier commands group definition. 2 total commands, 1 Subgroups, 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.wcdmaMeas.olpControl.carrier.clone()
```

Subgroups

6.4.2.1.1 UepPower

class UepPowerCls

UepPower commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.2.1.1.1 Rup<RampUpCarrier>

RepCap Settings

```
# Range: Nr1 .. Nr32
rc = driver.wcdmaMeas.olpControl.carrier.uepPower.rup.repcap_rampUpCarrier_get()
driver.wcdmaMeas.olpControl.carrier.uepPower.rup.repcap_rampUpCarrier_set(repcap.
↳ RampUpCarrier.Nr1)
```

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:OLPControl:CARRier<carrier>:UEPPower:RUP<rupcarrier>
FETCh:WCDMa:MEASurement<instance>:OLPControl:CARRier<carrier>:UEPPower:RUP<rupcarrier>
```

class RupCls

Rup commands group definition. 2 total commands, 0 Subgroups, 2 group commands Repeated Capability: RampUpCarrier, default value after init: RampUpCarrier.Nr1

fetch(cARRierExt=CARRierExt.Default, rampUpCarrier=RampUpCarrier.Default) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:OLPControl:CARRier<carrier>
↳ :UEPPower:RUP<rupcarrier>
value: List[float] = driver.wcdmaMeas.olpControl.carrier.uepPower.rup.
↳ fetch(cARRierExt = repcap.CARRierExt.Default, rampUpCarrier = repcap.
↳ RampUpCarrier.Default)
```

No command help available

Suppressed linked return values: reliability

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: No help available

read(cARRIERExt=CARRIERExt.Default, rampUpCarrier=RampUpCarrier.Default) → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:OLPControl:CARRIER<carrier>
↳:UEPPower:RUP<rupcarrier>
value: List[float] = driver.wcdmaMeas.olpControl.carrier.uepPower.rup.
↳read(cARRIERExt = repcap.CARRIERExt.Default, rampUpCarrier = repcap.
↳RampUpCarrier.Default)
```

No command help available

Suppressed linked return values: reliability

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: No help available

Cloning the Group

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

6.4.2.2 State

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:OLPControl:STATE
```

class StateCls

State commands group definition. 2 total commands, 1 Subgroups, 1 group commands

fetch(timeout: float = None, target_main_state: TargetMainState = None, target_sync_state: TargetSyncState = None) → ResourceState

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:OLPControl:STATE
value: enums.ResourceState = driver.wcdmaMeas.olpControl.state.fetch(timeout =
↳1.0, target_main_state = enums.TargetMainState.OFF, target_sync_state = enums.
↳TargetSyncState.ADJusted)
```

No command help available

param timeout

No help available

param target_main_state

No help available

```

param target_sync_state
    No help available

return
    state: No help available

```

Cloning the Group

```

# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.olpControl.state.clone()

```

Subgroups

6.4.2.2.1 All

SCPI Command :

```

FETCH:WCDMA:MEASurement<instance>:OLPControl:STATE:ALL

```

class AllCls

All commands group definition. 1 total commands, 0 Subgroups, 1 group commands

```

fetch(timeout: float = None, target_main_state: TargetMainState = None, target_sync_state: TargetSyncState
      = None) → List[ResourceState]

```

```

# SCPI: FETCH:WCDMA:MEASurement<instance>:OLPControl:STATE:ALL
value: List[enums.ResourceState] = driver.wcdmaMeas.olpControl.state.all.
↪ fetch(timeout = 1.0, target_main_state = enums.TargetMainState.OFF, target_
↪ sync_state = enums.TargetSyncState.ADJusted)

```

No command help available

```

param timeout
    No help available

param target_main_state
    No help available

param target_sync_state
    No help available

return
    state: No help available

```

6.4.3 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 OoSyncCls

OoSync commands group definition. 8 total commands, 1 Subgroups, 6 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: No parameter help available
- Out_Pow_Ab_Max: enums.ResultStatus2: No parameter help available
- Out_Pow_Ab_Min: enums.ResultStatus2: No parameter help available
- Out_Pow_Ccurrent: enums.ResultStatus2: No parameter help available
- Out_Pow_Cd_Max: enums.ResultStatus2: No parameter help available
- Out_Pow_Cd_Min: enums.ResultStatus2: No parameter help available
- Out_Pow_De_Max: enums.ResultStatus2: No parameter help available
- Out_Pow_De_Min: enums.ResultStatus2: No parameter help available
- Out_Pow_Fcurrent: enums.ResultStatus2: No parameter help available

class ResultData

Response structure. Fields:

- Reliability: int: No parameter help available
- Out_Pow_Ab_Max: float: No parameter help available
- Out_Pow_Ab_Min: float: No parameter help available
- Out_Pow_Ccurrent: float: No parameter help available
- Out_Powc_State: enums.OutPowFstate: No parameter help available
- Out_Pow_Cd_Max: float: No parameter help available
- Out_Pow_Cd_Min: float: No parameter help available
- Out_Pow_De_Max: float: No parameter help available
- Out_Pow_De_Min: float: No parameter help available
- Out_Pow_Fcurrent: float: No parameter help available
- Out_Pow_Fstate: enums.OutPowFstate: No parameter help available

abort(opc_timeout_ms: int = -1) → None

```
# SCPI: ABORt:WCDMa:MEASurement<instance>:OOSync
driver.wcdmaMeas.ooSync.abort()
```


No command help available

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:OOSync
value: CalculateStruct = driver.wcdmaMeas.ooSync.calculate()
```

No command help available

return

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

fetch() → ResultData

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:OOSync
value: ResultData = driver.wcdmaMeas.ooSync.fetch()
```

No command help available

return

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

initiate(opc_timeout_ms: int = -1) → None

```
# SCPI: INITiate:WCDMa:MEASurement<instance>:OOSync
driver.wcdmaMeas.ooSync.initiate()
```

No command help available

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

read() → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:OOSync
value: ResultData = driver.wcdmaMeas.ooSync.read()
```

No command help available

return

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

stop(opc_timeout_ms: int = -1) → None

```
# SCPI: STOP:WCDMa:MEASurement<instance>:OOSync
driver.wcdmaMeas.ooSync.stop()
```

No command help available

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

Cloning the Group

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

Subgroups

6.4.3.1 State

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:OOSync:STATe
```

class StateCls

State commands group definition. 2 total commands, 1 Subgroups, 1 group commands

fetch(timeout: float = None, target_main_state: TargetMainState = None, target_sync_state: TargetSyncState = None) → ResourceState

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:OOSync:STATe
value: enums.ResourceState = driver.wcdmaMeas.ooSync.state.fetch(timeout = 1.0,
↳ target_main_state = enums.TargetMainState.OFF, target_sync_state = enums.
↳ TargetSyncState.ADJusted)
```

No command help available

param timeout

No help available

param target_main_state

No help available

param target_sync_state

No help available

return

state: No help available

Cloning the Group

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

Subgroups

6.4.3.1.1 All

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:OOSync:STATe:ALL
```

class AllCls

All commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(timeout: float = None, target_main_state: TargetMainState = None, target_sync_state: TargetSyncState = None) → List[ResourceState]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:OOSync:STATe:ALL
value: List[enums.ResourceState] = driver.wcdmaMeas.ooSync.state.all.
↪ fetch(timeout = 1.0, target_main_state = enums.TargetMainState.OFF, target_
↪ sync_state = enums.TargetSyncState.ADJusted)
```

No command help available

param timeout

No help available

param target_main_state

No help available

param target_sync_state

No help available

return

state: No help available

6.4.4 Prach

SCPI Commands :

```
STOP:WCDMA:MEASurement<instance>:PRACH
ABORt:WCDMA:MEASurement<instance>:PRACH
INITiate:WCDMA:MEASurement<instance>:PRACH
```

class PrachCls

Prach commands group definition. 39 total commands, 4 Subgroups, 3 group commands

abort(opc_timeout_ms: int = -1) → None

```
# SCPI: ABORt:WCDMA:MEASurement<instance>:PRACH
driver.wcdmaMeas.prach.abort()
```

INTRO_CMD_HELP: Starts, stops **or** aborts the measurement:

- INITiate... starts **or** restarts the measurement. The measurement enters_

(continues on next page)

(continued from previous page)

↳ 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.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

initiate(opc_timeout_ms: int = -1) → None

```
# SCPI: INITiate:WCDMa:MEASurement<instance>:PRACH
driver.wcdmaMeas.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.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

stop(opc_timeout_ms: int = -1) → None

```
# SCPI: STOP:WCDMa:MEASurement<instance>:PRACH
driver.wcdmaMeas.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.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

Cloning the Group

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

Subgroups

6.4.4.1 OffPower

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:PRACH:OFFPower
READ:WCDMa:MEASurement<instance>:PRACH:OFFPower
CALCulate:WCDMa:MEASurement<instance>:PRACH:OFFPower
```

class OffPowerCls

OffPower commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate() → List[float]

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:PRACH:OFFPower
value: List[float or bool] = driver.wcdmaMeas.prach.offPower.calculate()
```

Return the OFF power results.

Suppressed linked return values: reliability

```
return
    off_power: (float or boolean items) OFF power before preamble, OFF power after
    preamble
```

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRACH:OFFPower
value: List[float] = driver.wcdmaMeas.prach.offPower.fetch()
```

Return the OFF power results.

Suppressed linked return values: reliability

```
return
    off_power: OFF power before preamble, OFF power after preamble
```

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:OFFPower
value: List[float] = driver.wcdmaMeas.prach.offPower.read()
```

Return the OFF power results.

Suppressed linked return values: reliability

```
return
    off_power: OFF power before preamble, OFF power after preamble
```

6.4.4.2 Preamble<Preamble>

RepCap Settings

```
# Range: Nr1 .. Nr5
rc = driver.wcdmaMeas.prach.preamble.repcap_preamble_get()
driver.wcdmaMeas.prach.preamble.repcap_preamble_set(repcap.Preamble.Nr1)
```

class PreambleCls

Preamble commands group definition. 3 total commands, 1 Subgroups, 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.wcdmaMeas.prach.preamble.clone()
```

Subgroups

6.4.4.2.1 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:PRACH:PREamble<nr>:CURRent
FETCh:WCDMa:MEASurement<instance>:PRACH:PREamble<nr>:CURRent
CALCulate:WCDMa:MEASurement<instance>:PRACH:PREamble<nr>:CURRent
```

class CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Ue_Power: float or bool: Mean preamble power
- Power_Steps: float or bool: Mean preamble power minus mean power of previous preamble For the first preamble, NCAP is returned.
- Carrier_Freq_Err: float or bool: Carrier frequency error
- Evm_Rms: float or bool: Error vector magnitude RMS value
- Evm_Peak: float or bool: Error vector magnitude peak value
- Mag_Error_Rms: float or bool: Magnitude error RMS value
- Mag_Error_Peak: float or bool: Magnitude error peak value
- Phase_Error_Rms: float or bool: No parameter help available
- Phase_Error_Peak: float or bool: No parameter help available

- Iq_Offset: float or bool: I/Q origin offset
- Iq_Imbalance: float or bool: I/Q imbalance

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Ue_Power: float: Mean preamble power
- Power_Steps: float: Mean preamble power minus mean power of previous preamble For the first preamble, NCAP is returned.
- Carrier_Freq_Err: float: Carrier frequency error
- Evm_Rms: float: Error vector magnitude RMS value
- Evm_Peak: float: Error vector magnitude peak value
- Mag_Error_Rms: float: Magnitude error RMS value
- Mag_Error_Peak: float: Magnitude error peak value
- Phase_Error_Rms: float: No parameter help available
- Phase_Error_Peak: float: No parameter help available
- Iq_Offset: float: I/Q origin offset
- Iq_Imbalance: float: I/Q imbalance
- Signature: int: Detected preamble signature

calculate(*preamble=Preamble.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:PRCh:PREamble<nr>:CURRent
value: CalculateStruct = driver.wcdmaMeas.prach.preamble.current.
↳ calculate(preamble = repcap.Preamble.Default)
```

Return the single value results for a selected preamble.

Suppressed linked return values: reliability

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*) → ResultData

```
# SCPI: FETCh:WCDma:MEASurement<instance>:PRCh:PREamble<nr>:CURRent
value: ResultData = driver.wcdmaMeas.prach.preamble.current.fetch(preamble =
↳ repcap.Preamble.Default)
```

Return the single value results for a selected preamble.

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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:PRACH:PREamble<nr>:CURRENT
value: ResultData = driver.wcdmaMeas.prach.preamble.current.read(preamble = ↵
↵repcap.Preamble.Default)
```

Return the single value results for a selected preamble.

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.

6.4.4.3 State

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:PRACH:STATe
```

class StateCls

State commands group definition. 2 total commands, 1 Subgroups, 1 group commands

fetch(*timeout: float = None, target_main_state: TargetMainState = None, target_sync_state: TargetSyncState = None*) → ResourceState

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:PRACH:STATe
value: enums.ResourceState = driver.wcdmaMeas.prach.state.fetch(timeout = 1.0, ↵
↵target_main_state = enums.TargetMainState.OFF, target_sync_state = enums.
↵TargetSyncState.ADJusted)
```

Queries the main measurement state. Without query parameters, the state is returned immediately. With query parameters, the state is returned when the <TargetMainState> and the <TargetSyncState> are reached or when the <Timeout> expires.

param timeout

No help available

param target_main_state

Target MainState for the query Default is RUN.

param target_sync_state

Target SyncState for the query Default is ADJ.

return

state: Current state or target state of ongoing state transition OFF: measurement off
 RUN: measurement running RDY: measurement completed

Cloning the Group

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

Subgroups

6.4.4.3.1 All

SCPI Command :

```
FETCH:WCDMA:MEASurement<instance>:PRACH:STATE:ALL
```

class AllCls

All commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(*timeout*: float = None, *target_main_state*: TargetMainState = None, *target_sync_state*: TargetSyncState = None) → List[ResourceState]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:PRACH:STATE:ALL
value: List[enums.ResourceState] = driver.wcdmaMeas.prach.state.all.
↪ fetch(timeout = 1.0, target_main_state = enums.TargetMainState.OFF, target_
↪ sync_state = enums.TargetSyncState.ADJusted)
```

Queries the main measurement state and the measurement substates. Without query parameters, the states are returned immediately. With query parameters, the states are returned when the <TargetMainState> and the <TargetSyncState> are reached or when the <Timeout> expires.

param timeout

No help available

param target_main_state

Target MainState for the query Default is RUN.

param target_sync_state

Target SyncState for the query Default is ADJ.

return

state: No help available

6.4.4.4 Trace

class TraceCls

Trace commands group definition. 28 total commands, 7 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.4.4.1 EvMagnitude

class EvMagnitudeCls

EvMagnitude commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.evMagnitude.clone()
```

Subgroups

6.4.4.4.1.1 Chip

class ChipCls

Chip commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.evMagnitude.chip.clone()
```

Subgroups

6.4.4.4.1.2 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:PRACH:TRACe:EvMagnitude:CHIP:CURRENT
FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:EvMagnitude:CHIP:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:EvMagnitude:CHIP:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.evMagnitude.chip.current.
    ↪ fetch()
```

Return the values of the error vector magnitude vs chip diagram.

Suppressed linked return values: reliability

return

evm_chip: Comma-separated list of 4096 values, one per chip of the preselected preamble

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:PRACH:TRAcE:EVMagnitude:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.evMagnitude.chip.current.
↳ read()
```

Return the values of the error vector magnitude vs chip diagram.

Suppressed linked return values: reliability

return

evm_chip: Comma-separated list of 4096 values, one per chip of the preselected preamble

6.4.4.4.1.3 Peak

class PeakCls

Peak commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.evMagnitude.peak.clone()
```

Subgroups

6.4.4.4.1.4 Current

SCPI Commands :

```
FETCH:WCDma:MEASurement<instance>:PRACH:TRAcE:EVMagnitude:PEAK:CURRent
READ:WCDma:MEASurement<instance>:PRACH:TRAcE:EVMagnitude:PEAK:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDma:MEASurement<instance>:PRACH:TRAcE:EVMagnitude:PEAK:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.evMagnitude.peak.current.
↳ fetch()
```

Return the EVM RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

evm: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude:PEAK:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.evMagnitude.peak.current.
↪ read()
```

Return the EVM RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

evm: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

6.4.4.4.1.5 Rms

class RmsCls

Rms commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.evMagnitude.rms.clone()
```

Subgroups

6.4.4.4.1.6 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude[:RMS]:CURRENT
FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude[:RMS]:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude[:RMS]:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.evMagnitude.rms.current.
↪ fetch()
```

Return the EVM RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

evm: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:PRACH:TRACE:EVMagnitude[:RMS]:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.evMagnitude.rms.current.read()
```

Return the EVM RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

evm: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

6.4.4.4.2 FreqError

class FreqErrorCls

FreqError commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.freqError.clone()
```

Subgroups

6.4.4.4.2.1 Current

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:PRACH:TRACE:FERRor:CURRENT
FETCH:WCDMA:MEASurement<instance>:PRACH:TRACE:FERRor:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:PRACH:TRACE:FERRor:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.freqError.current.fetch()
```

Return the bargraph values of the frequency error.

Suppressed linked return values: reliability

return

frequency_error: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:PRACH:TRACE:FERRor:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.freqError.current.read()
```

Return the bargraph values of the frequency error.

Suppressed linked return values: reliability

return

frequency_error: Comma-separated list of values, one result per measured preamble
(see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

6.4.4.4.3 Iq

class IqCls

Iq commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.iq.clone()
```

Subgroups

6.4.4.4.3.1 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:PRACH:TRACe:IQ:CURRent
FETCh:WCDMa:MEASurement<instance>:PRACH:TRACe:IQ:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Iphase: List[float]: I amplitude of a constellation point
- Qphase: List[float]: Q amplitude of a constellation point

fetch() → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:PRACH:TRACe:IQ:CURRent
value: ResultData = driver.wcdmaMeas.prach.trace.iq.current.fetch()
```

Returns the results in the 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.wcdmaMeas.prach.trace.iq.current.read()
```

Returns the results in the 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.

6.4.4.4.4 Merror

class MerrorCls

Merror commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.merror.clone()
```

Subgroups

6.4.4.4.4.1 Chip

class ChipCls

Chip commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.merror.chip.clone()
```

Subgroups

6.4.4.4.4.2 Current

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:MERRor:CHIP:CURRent
READ:WCDMa:MEASurement<instance>:PRACH:TRACe:MERRor:CHIP:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:MERRor:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.merror.chip.current.fetch()
```

Return the values of the magnitude error vs chip diagram.

Suppressed linked return values: reliability

```
        return
            mag_error_chip: Comma-separated list of 4096 values, one per chip of the preselected
            preamble

read() → List[float]
```

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRCh:TRAcE:MERRor:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.merror.chip.current.read()
```

Return the values of the magnitude error vs chip diagram.

Suppressed linked return values: reliability

```
        return
            mag_error_chip: Comma-separated list of 4096 values, one per chip of the preselected
            preamble
```

6.4.4.4.4.3 Peak

class PeakCls

Peak commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.merror.peak.clone()
```

Subgroups

6.4.4.4.4.4 Current

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:PRCh:TRAcE:MERRor:PEAK:CURRent
READ:WCDMa:MEASurement<instance>:PRCh:TRAcE:MERRor:PEAK:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

```
fetch() → List[float]
```

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRCh:TRAcE:MERRor:PEAK:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.merror.peak.current.fetch()
```

Return the magnitude error RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

```
        return
            magnitude_error: Comma-separated list of values, one result per measured preamble
            (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)
```


read() → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:PRACH:TRACE:MERROR:PEAK:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.merror.peak.current.read()
```

Return the magnitude error RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

magnitude_error: Comma-separated list of values, one result per measured preamble
(see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

6.4.4.4.5 Rms

class RmsCls

Rms commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.merror.rms.clone()
```

Subgroups

6.4.4.4.6 Current

SCPI Commands :

```
FETCH:WCDMA:MEASUREMENT<instance>:PRACH:TRACE:MERROR[:RMS]:CURRENT
READ:WCDMA:MEASUREMENT<instance>:PRACH:TRACE:MERROR[:RMS]:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:PRACH:TRACE:MERROR[:RMS]:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.merror.rms.current.fetch()
```

Return the magnitude error RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

magnitude_error: Comma-separated list of values, one result per measured preamble
(see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

read() → List[float]

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:PRACH:TRACE:MERROR[:RMS]:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.merror.rms.current.read()
```

Return the magnitude error RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

```
return
    magnitude_error: Comma-separated list of values, one result per measured preamble
    (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)
```

6.4.4.4.5 Perror

class PerrorCls

Perror commands group definition. 6 total commands, 3 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.perror.clone()
```

Subgroups

6.4.4.4.5.1 Chip

class ChipCls

Chip commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.perror.chip.clone()
```

Subgroups

6.4.4.4.5.2 Current

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:PERRor:CHIP:CURRent
READ:WCDMa:MEASurement<instance>:PRACH:TRACe:PERRor:CHIP:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRACH:TRACe:PERRor:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.perror.chip.current.fetch()
```

Return the values of the phase error vs chip diagram.

Suppressed linked return values: reliability

return

phase_error_chip: Comma-separated list of 4096 values, one per chip of the preselected preamble

read() → List[float]

```
# SCPI: READ:WCDma:MEASurement<instance>:PRCh:TRAcE:PERRor:CHIP:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.perror.chip.current.read()
```

Return the values of the phase error vs chip diagram.

Suppressed linked return values: reliability

return

phase_error_chip: Comma-separated list of 4096 values, one per chip of the preselected preamble

6.4.4.4.5.3 Peak

class PeakCls

Peak commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.perror.peak.clone()
```

Subgroups

6.4.4.4.5.4 Current

SCPI Commands :

```
FETCh:WCDma:MEASurement<instance>:PRCh:TRAcE:PERRor:PEAK:CURRent
READ:WCDma:MEASurement<instance>:PRCh:TRAcE:PERRor:PEAK:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDma:MEASurement<instance>:PRCh:TRAcE:PERRor:PEAK:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.perror.peak.current.fetch()
```

Return the phase error RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

phase_error: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRCh:TRAcE:PERRor:PEAK:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.perror.peak.current.read()
```

Return the phase error RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

phase_error: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

6.4.4.4.5.5 Rms

class RmsCls

Rms commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.perror.rms.clone()
```

Subgroups

6.4.4.4.5.6 Current

SCPI Commands :

```
FETCH:WCDMa:MEASurement<instance>:PRCh:TRAcE:PERRor[:RMS]:CURRent
READ:WCDMa:MEASurement<instance>:PRCh:TRAcE:PERRor[:RMS]:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMa:MEASurement<instance>:PRCh:TRAcE:PERRor[:RMS]:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.perror.rms.current.fetch()
```

Return the phase error RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

phase_error: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:PRACH:TRACe:PERRor[:RMS]:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.perror.rms.current.read()
```

Return the phase error RMS and peak values for each measured preamble.

Suppressed linked return values: reliability

return

phase_error: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

6.4.4.4.6 Psteps

class PstepsCls

Psteps commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.psteps.clone()
```

Subgroups

6.4.4.4.6.1 Current

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:PRACH:TRACe:PSTeps:CURRent
FETCH:WCDMA:MEASurement<instance>:PRACH:TRACe:PSTeps:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:PRACH:TRACe:PSTeps:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.psteps.current.fetch()
```

Return the bargraph values of the power steps.

Suppressed linked return values: reliability

return

power_steps: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble) For the first preamble, NCAP is returned.

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:PRCh:TRAcE:PSTeps:CURRent
value: List[float] = driver.wcdmaMeas.prach.trace.psteps.current.read()
```

Return the bargraph values of the power steps.

Suppressed linked return values: reliability

return

power_steps: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble) For the first preamble, NCAP is returned.

6.4.4.4.7 UePower

class UePowerCls

UePower commands group definition. 4 total commands, 2 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.uePower.clone()
```

Subgroups

6.4.4.4.7.1 Chip

class ChipCls

Chip commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.prach.trace.uePower.chip.clone()
```

Subgroups

6.4.4.4.7.2 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:PRCh:TRAcE:UEPower:CHIP:CURRent
FETCh:WCDMa:MEASurement<instance>:PRCh:TRAcE:UEPower:CHIP:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:PRACH:TRACE:UEPower:CHIP:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.uePower.chip.current.fetch()
```

Return the values of the UE power vs chip diagram.

Suppressed linked return values: reliability

return

ue_power_chip: Comma-separated list of 9216 values, one per chip: 2560 values before last preamble, 4096 values for preselected preamble, 2560 values after last preamble

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:PRACH:TRACE:UEPower:CHIP:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.uePower.chip.current.read()
```

Return the values of the UE power vs chip diagram.

Suppressed linked return values: reliability

return

ue_power_chip: Comma-separated list of 9216 values, one per chip: 2560 values before last preamble, 4096 values for preselected preamble, 2560 values after last preamble

6.4.4.4.7.3 Current

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:PRACH:TRACE:UEPower:CURRENT
FETCH:WCDMA:MEASurement<instance>:PRACH:TRACE:UEPower:CURRENT
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:PRACH:TRACE:UEPower:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.uePower.current.fetch()
```

Return the result values of the UE power bargraph.

Suppressed linked return values: reliability

return

ue_power: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

read() → List[float]

```
# SCPI: READ:WCDMA:MEASurement<instance>:PRACH:TRACE:UEPower:CURRENT
value: List[float] = driver.wcdmaMeas.prach.trace.uePower.current.read()
```

Return the result values of the UE power bargraph.

Suppressed linked return values: reliability

return

ue_power: Comma-separated list of values, one result per measured preamble (see method RsCMPX_WcdmaMeas.Configure.WcdmaMeas.Prach.mpreamble)

6.4.5 Tpc

SCPI Commands :

```
STOP:WCDMa:MEASurement<instance>:TPC
ABORT:WCDMa:MEASurement<instance>:TPC
INITiate:WCDMa:MEASurement<instance>:TPC
```

class TpcCls

Tpc commands group definition. 54 total commands, 4 Subgroups, 3 group commands

abort(opc_timeout_ms: int = -1) → None

```
# SCPI: ABORT:WCDMa:MEASurement<instance>:TPC
driver.wcdmaMeas.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.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

initiate(opc_timeout_ms: int = -1) → None

```
# SCPI: INITiate:WCDMa:MEASurement<instance>:TPC
driver.wcdmaMeas.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.

(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.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

stop(opc_timeout_ms: int = -1) → None

```
# SCPI: STOP:WCDMA:MEASurement<instance>:TPC
driver.wcdmaMeas.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.

param opc_timeout_ms

Maximum time to wait in milliseconds, valid only for this call.

Cloning the Group

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

Subgroups

6.4.5.1 Carrier<Carrier>

RepCap Settings

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

class CarrierCls

Carrier commands group definition. 28 total commands, 3 Subgroups, 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.wcdmaMeas.tpc.carrier.clone()
```

Subgroups

6.4.5.1.1 Psteps

class PstepsCls

Psteps commands group definition. 11 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.5.1.1.1 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 AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Pwr_Steps_0_Db: float or bool: No parameter help available
- Pwr_Steps_B_1_Db: float or bool: No parameter help available
- Pwr_Steps_Cm_1_Db: float or bool: No parameter help available
- Pwr_Steps_Group_A: float or bool: No parameter help available
- Pwr_Steps_Group_B: float or bool: No parameter help available
- Pwr_Steps_Group_C: float or bool: No parameter help available
- Start_Slot_Group_A: float or bool: No parameter help available
- Pwr_Steps_Eg: float or bool: No parameter help available
- Pwr_Steps_Fh: float or bool: No parameter help available
- Pwr_Steps_Group_Eg: float or bool: No parameter help available

- Pwr_Steps_Group_Fh: float or bool: No parameter help available
- Start_Slot_Group_Eg: float or bool: No parameter help available
- Start_Slot_Group_Fh: float or bool: No parameter help available
- Pwr_Steps_Up: float or bool: No parameter help available
- Pwr_Steps_Down: float or bool: 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: '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*) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:PSTeps:AVERage
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:AVERage
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:AVERage
value: ResultData = driver.wcdmaMeas.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.

6.4.5.1.1.2 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 MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Pwr_Steps_0_Db: float or bool: No parameter help available
- Pwr_Steps_B_1_Db: float or bool: No parameter help available
- Pwr_Steps_Cm_1_Db: float or bool: No parameter help available
- Pwr_Steps_Group_A: float or bool: No parameter help available
- Pwr_Steps_Group_B: float or bool: No parameter help available
- Pwr_Steps_Group_C: float or bool: No parameter help available
- Start_Slot_Group_A: float or bool: No parameter help available
- Pwr_Steps_Eg: float or bool: No parameter help available
- Pwr_Steps_Fh: float or bool: No parameter help available
- Pwr_Steps_Group_Eg: float or bool: No parameter help available
- Pwr_Steps_Group_Fh: float or bool: No parameter help available
- Start_Slot_Group_Eg: float or bool: No parameter help available
- Start_Slot_Group_Fh: float or bool: No parameter help available
- Pwr_Steps_Up: float or bool: No parameter help available
- Pwr_Steps_Down: float or bool: 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: '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*) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:TPC:CARRier<carrier>
↪:PSTeps:MAXimum
value: CalculateStruct = driver.wcdmaMeas.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. 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*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:MAXimum
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:MAXimum
value: ResultData = driver.wcdmaMeas.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.

6.4.5.1.1.3 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 MinimumCls

Minimum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Pwr_Steps_0_Db: float or bool: No parameter help available
- Pwr_Steps_B_1_Db: float or bool: No parameter help available
- Pwr_Steps_Cm_1_Db: float or bool: No parameter help available
- Pwr_Steps_Group_A: float or bool: No parameter help available
- Pwr_Steps_Group_B: float or bool: No parameter help available
- Pwr_Steps_Group_C: float or bool: No parameter help available
- Start_Slot_Group_A: float or bool: No parameter help available
- Pwr_Steps_Eg: float or bool: No parameter help available
- Pwr_Steps_Fh: float or bool: No parameter help available
- Pwr_Steps_Group_Eg: float or bool: No parameter help available
- Pwr_Steps_Group_Fh: float or bool: No parameter help available
- Start_Slot_Group_Eg: float or bool: No parameter help available
- Start_Slot_Group_Fh: float or bool: No parameter help available
- Pwr_Steps_Up: float or bool: No parameter help available
- Pwr_Steps_Down: float or bool: 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: '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*) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
→:PSTeps:MINimum
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:MINimum
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:MINimum
value: ResultData = driver.wcdmaMeas.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.

6.4.5.1.1.4 Statistics

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:STATistics
FETCH:WCDMA:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:STATistics
```

class StatisticsCls

Statistics commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: ‘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: Power steps up result of Change of TFC mode.
- Pwr_Steps_Down: int: Power steps down result of Change of TFC mode.
- Rpwr_Steps: int: 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*) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:STATistics
value: ResultData = driver.wcdmaMeas.tpc.carrier.psteps.statistics.
↪ fetch(carrier = repcap.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*) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:PSTeps:STATistics
value: ResultData = driver.wcdmaMeas.tpc.carrier.psteps.statistics.read(carrier,
↪ repcap.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.

6.4.5.1.2 Trace

class TraceCls

Trace commands group definition. 6 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.5.1.2.1 Psteps

class PstepsCls

Psteps commands group definition. 3 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.5.1.2.2 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 CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(carrier=Carrier.Default) → List[float]

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↪:TRACe:PSTeps:CURRent
value: List[float or bool] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

`power_steps`: (float or boolean items) `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`.

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳ :TRACe:PSTeps:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

`power_steps`: `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`.

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳ :TRACe:PSTeps:CURRent
value: List[float] = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

param carrier

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

return

power_steps: 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.

6.4.5.1.2.3 UePower

class UePowerCls

UePower commands group definition. 3 total commands, 1 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.5.1.2.4 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 CurrentCls

Current commands group definition. 3 total commands, 0 Subgroups, 3 group commands

calculate(carrier=Carrier.Default) → List[float]

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↪:TRACe:UEPower:CURRent
value: List[float or bool] = driver.wcdmaMeas.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:...:MLENgtH? 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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: (float or boolean items) N power results, one per measured slot

fetch(*carrier=Carrier.Default*) → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↪:TRACe:UEPower:CURRent
value: List[float] = driver.wcdmaMeas.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:...:MLENgtH? 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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: N power results, one per measured slot

read(*carrier=Carrier.Default*) → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↪:TRACe:UEPower:CURRent
value: List[float] = driver.wcdmaMeas.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:...:MLENgtH? 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.

Suppressed linked return values: reliability

param carrier

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

return

ue_power: N power results, one per measured slot

6.4.5.1.3 UePower

class UePowerCls

UePower commands group definition. 11 total commands, 4 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.5.1.3.1 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 AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Ue_Power: float or bool: UE power
- Max_Output_Power: float or bool: Maximum output power
- Min_Outpu_Power: float or bool: Minimum output power

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Ue_Power: float: UE power
- Max_Output_Power: float: Maximum output power
- Min_Outpu_Power: float: Minimum output power

calculate(carrier=*Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:UEPower:AVERage
value: CalculateStruct = driver.wcdmaMeas.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) → ResultData

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:AVERage
value: ResultData = driver.wcdmaMeas.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) → ResultData

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:AVERage
value: ResultData = driver.wcdmaMeas.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.

6.4.5.1.3.2 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 MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Ue_Power: float or bool: UE power
- Max_Output_Power: float or bool: Maximum output power
- Min_Outpu_Power: float or bool: Minimum output power

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Ue_Power: float: UE power
- Max_Output_Power: float: Maximum output power
- Min_Outpu_Power: float: Minimum output power

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:TPC:CARRier<carrier>
↳:UEPower:MAXimum
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCh:WCDma:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:MAXimum
value: ResultData = driver.wcdmaMeas.tpc.carrier.uePower.maximum.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*) → ResultData

```
# SCPI: READ:WCDMA:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:MAXimum
value: ResultData = driver.wcdmaMeas.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.

6.4.5.1.3.3 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 MinimumCls

Minimum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: ‘Reliability indicator’
- Ue_Power: float or bool: UE power
- Max_Output_Power: float or bool: Maximum output power
- Min_Outpu_Power: float or bool: Minimum output power

class ResultData

Response structure. Fields:

- Reliability: int: ‘Reliability indicator’
- Ue_Power: float: UE power
- Max_Output_Power: float: Maximum output power
- Min_Outpu_Power: float: Minimum output power

calculate(*carrier=Carrier.Default*) → CalculateStruct

```
# SCPI: CALCulate:WCDma:MEASurement<instance>:TPC:CARRier<carrier>
↳:UEPower:MINimum
value: CalculateStruct = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: FETCh:WCDma:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:MINimum
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```
# SCPI: READ:WCDma:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:MINimum
value: ResultData = driver.wcdmaMeas.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.

6.4.5.1.3.4 Statistics

SCPI Commands :

```

FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:STATistics
READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:STATistics

```

class StatisticsCls

Statistics commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Max_Output_Power: int: Number of trace values for maximum output power
- Min_Outpu_Power: int: Number of trace values for minimum output power

fetch(*carrier=Carrier.Default*) → ResultData

```

# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>
↳:UEPower:STATistics
value: ResultData = driver.wcdmaMeas.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*) → ResultData

```

# SCPI: READ:WCDMa:MEASurement<instance>:TPC:CARRier<carrier>:UEPower:STATistics
value: ResultData = driver.wcdmaMeas.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.

6.4.5.2 Dhib

class DhibCls

Dhib commands group definition. 11 total commands, 5 Subgroups, 0 group commands

Cloning the Group

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

Subgroups

6.4.5.2.1 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 AverageCls

Average commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Carrier_Ch_Power: float or bool: Level of the uplink carrier, where the UE transmits at the maximal output power.
- Inband_Emission: float or bool: Relative level of the other uplink carrier transmitting at minimal output power.

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Carrier_Ch_Power: float: Level of the uplink carrier, where the UE transmits at the maximal output power.
- Inband_Emission: float: Relative level of the other uplink carrier transmitting at minimal output power.

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:DHIB:AVERage
value: CalculateStruct = driver.wcdmaMeas.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.wcdmaMeas.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.wcdmaMeas.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.

6.4.5.2.2 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 MaximumCls

Maximum commands group definition. 3 total commands, 0 Subgroups, 3 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Carrier_Ch_Power: float or bool: Level of the uplink carrier, where the UE transmits at the maximal output power.
- Inband_Emission: float or bool: Relative level of the other uplink carrier transmitting at minimal output power.

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'

- Carrier_Ch_Power: float: Level of the uplink carrier, where the UE transmits at the maximal output power.
- Inband_Emission: float: Relative level of the other uplink carrier transmitting at minimal output power.

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMa:MEASurement<instance>:TPC:DHIB:MAXimum
value: CalculateStruct = driver.wcdmaMeas.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.wcdmaMeas.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.wcdmaMeas.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.

6.4.5.2.3 Minimum

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:TPC:DHIB:MINimum
FETCH:WCDMa:MEASurement<instance>:TPC:DHIB:MINimum
```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'

- Carrier_Ch_Power: float: Level of the uplink carrier, where the UE transmits at the maximal output power.
- Inband_Emission: float: Relative level of the other uplink carrier transmitting at minimal output power.

fetch() → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:TPC:DHIB:MINimum
value: ResultData = driver.wcdmaMeas.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.wcdmaMeas.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.

6.4.5.2.4 Minimumc

SCPI Command :

```
CALCulate:WCDMA:MEASurement<instance>:TPC:DHIB:MINimumc
```

class MinimumcCls

Minimumc commands group definition. 1 total commands, 0 Subgroups, 1 group commands

class CalculateStruct

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Carrier_Ch_Power: float or bool: Level of the uplink carrier, where the UE transmits at the maximal output power.
- Inband_Emission: float or bool: Relative level of the other uplink carrier transmitting at minimal output power.

calculate() → CalculateStruct

```
# SCPI: CALCulate:WCDMA:MEASurement<instance>:TPC:DHIB:MINimumc
value: CalculateStruct = driver.wcdmaMeas.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.

6.4.5.2.5 Statistics

SCPI Commands :

```
READ:WCDMA:MEASurement<instance>:TPC:DHIB:STATistics
FETCh:WCDMA:MEASurement<instance>:TPC:DHIB:STATistics
```

class StatisticsCls

Statistics commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → float

```
# SCPI: FETCh:WCDMA:MEASurement<instance>:TPC:DHIB:STATistics
value: float = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

statistics: No help available

read() → float

```
# SCPI: READ:WCDMA:MEASurement<instance>:TPC:DHIB:STATistics
value: float = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

statistics: No help available

6.4.5.3 State

SCPI Command :

```
FETCh:WCDMA:MEASurement<instance>:TPC:STATE
```

class StateCls

State commands group definition. 2 total commands, 1 Subgroups, 1 group commands

fetch(*timeout*: float = None, *target_main_state*: TargetMainState = None, *target_sync_state*: TargetSyncState = None) → ResourceState

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:STATE
value: enums.ResourceState = driver.wcdmaMeas.tpc.state.fetch(timeout = 1.0,
↳ target_main_state = enums.TargetMainState.OFF, target_sync_state = enums.
↳ TargetSyncState.ADJusted)
```

Queries the main measurement state. Without query parameters, the state is returned immediately. With query parameters, the state is returned when the <TargetMainState> and the <TargetSyncState> are reached or when the <Timeout> expires.

param timeout

No help available

param target_main_state

Target MainState for the query Default is RUN.

param target_sync_state

Target SyncState for the query Default is ADJ.

return

state: Current state or target state of ongoing state transition OFF: measurement off
RUN: measurement running RDY: measurement completed

Cloning the Group

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

Subgroups

6.4.5.3.1 All

SCPI Command :

```
FETCh:WCDMa:MEASurement<instance>:TPC:STATE:ALL
```

class AllCls

All commands group definition. 1 total commands, 0 Subgroups, 1 group commands

fetch(timeout: float = None, target_main_state: TargetMainState = None, target_sync_state: TargetSyncState = None) → List[ResourceState]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:STATE:ALL
value: List[enums.ResourceState] = driver.wcdmaMeas.tpc.state.all.fetch(timeout=
↳ 1.0, target_main_state = enums.TargetMainState.OFF, target_sync_state =
↳ enums.TargetSyncState.ADJusted)
```

Queries the main measurement state and the measurement substates. Without query parameters, the states are returned immediately. With query parameters, the states are returned when the <TargetMainState> and the <TargetSyncState> are reached or when the <Timeout> expires.

param timeout

No help available

param target_main_state

Target MainState for the query Default is RUN.

param target_sync_state

Target SyncState for the query Default is ADJ.

return

state: No help available

6.4.5.4 Total**class TotalCls**

Total commands group definition. 10 total commands, 2 Subgroups, 0 group commands

Cloning the Group

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

Subgroups**6.4.5.4.1 Trace****class TraceCls**

Trace commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.tpc.total.trace.clone()
```

Subgroups**6.4.5.4.1.1 UePower****class UePowerCls**

UePower commands group definition. 2 total commands, 1 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.tpc.total.trace.uePower.clone()
```

Subgroups

6.4.5.4.1.2 Current

SCPI Commands :

```
READ:WCDMa:MEASurement<instance>:TPC:TOTal:TRACe:UEPower:CURRent
FETCh:WCDMa:MEASurement<instance>:TPC:TOTal:TRACe:UEPower:CURRent
```

class CurrentCls

Current commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → List[float]

```
# SCPI: FETCh:WCDMa:MEASurement<instance>:TPC:TOTal:TRACe:UEPower:CURRent
value: List[float] = driver.wcdmaMeas.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:...:MLENgt? command of the used measurement mode.

Suppressed linked return values: reliability

return

ue_power: N power results, one per measured slot

read() → List[float]

```
# SCPI: READ:WCDMa:MEASurement<instance>:TPC:TOTal:TRACe:UEPower:CURRent
value: List[float] = driver.wcdmaMeas.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:...:MLENgt? command of the used measurement mode.

Suppressed linked return values: reliability

return

ue_power: N power results, one per measured slot

6.4.5.4.2 UePower

class UePowerCls

UePower commands group definition. 8 total commands, 4 Subgroups, 0 group commands

Cloning the Group

```
# Create a clone of the original group, that exists independently
group2 = driver.wcdmaMeas.tpc.total.uePower.clone()
```

Subgroups

6.4.5.4.2.1 Average

SCPI Commands :

```
FETCH:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:AVERage
READ:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:AVERage
```

class AverageCls

Average commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Ue_Power: float: UE power
- Max_Output_Power: float: Maximum output power

fetch() → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:AVERage
value: ResultData = driver.wcdmaMeas.tpc.total.uePower.average.fetch()
```

Return the UE power and the 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.wcdmaMeas.tpc.total.uePower.average.read()
```

Return the UE power and the 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.

6.4.5.4.2.2 Maximum

SCPI Commands :

```
FETCH:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MAXimum
READ:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MAXimum
```

class MaximumCls

Maximum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Ue_Power: float: UE power
- Max_Output_Power: float: Maximum output power

fetch() → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MAXimum
value: ResultData = driver.wcdmaMeas.tpc.total.uePower.maximum.fetch()
```

Return the UE power and the 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.wcdmaMeas.tpc.total.uePower.maximum.read()
```

Return the UE power and the 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.

6.4.5.4.2.3 Minimum**SCPI Commands :**

```
FETCH:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MINimum
READ:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MINimum
```

class MinimumCls

Minimum commands group definition. 2 total commands, 0 Subgroups, 2 group commands

class ResultData

Response structure. Fields:

- Reliability: int: 'Reliability indicator'
- Ue_Power: float: UE power
- Max_Output_Power: float: Maximum output power

fetch() → ResultData

```
# SCPI: FETCH:WCDMA:MEASurement<instance>:TPC:TOTal:UEPower:MINimum
value: ResultData = driver.wcdmaMeas.tpc.total.uePower.minimum.fetch()
```

Return the UE power and the 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.wcdmaMeas.tpc.total.uePower.minimum.read()
```

Return the UE power and the 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.

6.4.5.4.2.4 Statistics

SCPI Commands :

```
FETCH:WCDMA:MEASUREMENT<instance>:TPC:TOTAL:UEPower:STATISTICS
READ:WCDMA:MEASUREMENT<instance>:TPC:TOTAL:UEPower:STATISTICS
```

class StatisticsCls

Statistics commands group definition. 2 total commands, 0 Subgroups, 2 group commands

fetch() → int

```
# SCPI: FETCH:WCDMA:MEASUREMENT<instance>:TPC:TOTAL:UEPower:STATISTICS
value: int = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

max_output_power: Number of trace values for maximum output power over all carriers

read() → int

```
# SCPI: READ:WCDMA:MEASUREMENT<instance>:TPC:TOTAL:UEPower:STATISTICS
value: int = driver.wcdmaMeas.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.

Suppressed linked return values: reliability

return

max_output_power: Number of trace values for maximum output power over all carriers

RSCMPX_WCDMAMEAS UTILITIES

class Utilities

Common utility class. Utility functions common for all types of drivers.

Access snippet: `utils = RsCMPX_WcdmaMeas.utilities`

property logger: *ScpiLogger*

Scpi Logger interface, see [here](#)

Access snippet: `logger = RsCMPX_WcdmaMeas.utilities.logger`

property driver_version: `str`

Returns the instrument driver version.

property idn_string: `str`

Returns instrument's identification string - the response on the SCPI command `*IDN?`

property manufacturer: `str`

Returns manufacturer of the instrument.

property full_instrument_model_name: `str`

Returns the current instrument's full name e.g. 'FSW26'.

property instrument_model_name: `str`

Returns the current instrument's family name e.g. 'FSW'.

property supported_models: `List[str]`

Returns a list of the instrument models supported by this instrument driver.

property instrument_firmware_version: `str`

Returns instrument's firmware version.

property instrument_serial_number: `str`

Returns instrument's serial_number.

query_opc(*timeout: int = 0*) → `int`

SCPI command: `*OPC?` Queries the instrument's OPC bit and hence it waits until the instrument reports operation complete. If you define `timeout > 0`, the VISA timeout is set to that value just for this method call.

property instrument_status_checking: `bool`

Sets / returns Instrument Status Checking. When True (default is True), all the driver methods and properties are sending "SYSTem:ERRor?" at the end to immediately react on error that might have occurred. We recommend to keep the state checking ON all the time. Switch it OFF only in rare cases when you require maximum speed. The default state after initializing the session is ON.

property encoding: str

Returns string<=>bytes encoding of the session.

property opc_query_after_write: bool

Sets / returns Instrument *OPC? query sending after each command write. When True, (default is False) the driver sends *OPC? every time a write command is performed. Use this if you want to make sure your sequence is performed command-after-command.

property bin_float_numbers_format: BinFloatFormat

Sets / returns format of float numbers when transferred as binary data.

property bin_int_numbers_format: BinIntFormat

Sets / returns format of integer numbers when transferred as binary data.

clear_status() → None

Clears instrument's status system, the session's I/O buffers and the instrument's error queue.

query_all_errors() → List[str]

Queries and clears all the errors from the instrument's error queue. The method returns list of strings as error messages. If no error is detected, the return value is None. The process is: querying 'SYS-Tem:ERror?' in a loop until the error queue is empty. If you want to include the error codes, call the query_all_errors_with_codes()

query_all_errors_with_codes() → List[Tuple[int, str]]

Queries and clears all the errors from the instrument's error queue. The method returns list of tuples (code: int, message: str). If no error is detected, the return value is None. The process is: querying 'SYS-Tem:ERror?' in a loop until the error queue is empty.

property instrument_options: List[str]

Returns all the instrument options. The options are sorted in the ascending order starting with K-options and continuing with B-options.

reset() → None

SCPI command: *RST Sends *RST command + calls the clear_status().

default_instrument_setup() → None

Custom steps performed at the init and at the reset().

self_test(timeout: int = None) → Tuple[int, str]

SCPI command: *TST? Performs instrument's self-test. Returns tuple (code:int, message: str). Code 0 means the self-test passed. You can define the custom timeout in milliseconds. If you do not define it, the default selftest timeout is used (usually 60 secs).

is_connection_active() → bool

Returns true, if the VISA connection is active and the communication with the instrument still works.

reconnect(force_close: bool = False) → bool

If the connection is not active, the method tries to reconnect to the device. If the connection is active, and force_close is False, the method does nothing. If the connection is active, and force_close is True, the method closes, and opens the session again. Returns True, if the reconnection has been performed.

property resource_name: int

Returns the resource name used in the constructor

property opc_timeout: int

Sets / returns timeout in milliseconds for all the operations that use OPC synchronization.

property visa_timeout: int

Sets / returns visa IO timeout in milliseconds.

property data_chunk_size: int

Sets / returns the maximum size of one block transferred during write/read operations

property visa_manufacturer: int

Returns the manufacturer of the current VISA session.

process_all_commands() → None

SCPI command: ***WAI** Stops further commands processing until all commands sent before ***WAI** have been executed.

write_str(cmd: str) → None

Writes the command to the instrument.

write(cmd: str) → None

This method is an alias to the write_str(). Writes the command to the instrument as string.

write_int(cmd: str, param: int) → None

Writes the command to the instrument followed by the integer parameter: e.g.: cmd = 'SELECT:INPUT' param = '2', result command = 'SELECT:INPUT 2'

write_int_with_opc(cmd: str, param: int, timeout: int = None) → None

Writes the command with OPC to the instrument followed by the integer parameter: e.g.: cmd = 'SELECT:INPUT' param = '2', result command = 'SELECT:INPUT 2' If you do not provide timeout, the method uses current opc_timeout.

write_float(cmd: str, param: float) → None

Writes the command to the instrument followed by the boolean parameter: e.g.: cmd = 'CENTER:FREQ' param = '10E6', result command = 'CENTER:FREQ 10E6'

write_float_with_opc(cmd: str, param: float, timeout: int = None) → None

Writes the command with OPC to the instrument followed by the boolean parameter: e.g.: cmd = 'CENTER:FREQ' param = '10E6', result command = 'CENTER:FREQ 10E6' If you do not provide timeout, the method uses current opc_timeout.

write_bool(cmd: str, param: bool) → None

Writes the command to the instrument followed by the boolean parameter: e.g.: cmd = 'OUTPUT' param = 'True', result command = 'OUTPUT ON'

write_bool_with_opc(cmd: str, param: bool, timeout: int = None) → None

Writes the command with OPC to the instrument followed by the boolean parameter: e.g.: cmd = 'OUTPUT' param = 'True', result command = 'OUTPUT ON' If you do not provide timeout, the method uses current opc_timeout.

query_str(query: str) → str

Sends the query to the instrument and returns the response as string. The response is trimmed of any trailing LF characters and has no length limit.

query(query: str) → str

This method is an alias to the query_str(). Sends the query to the instrument and returns the response as string. The response is trimmed of any trailing LF characters and has no length limit.

query_bool(query: str) → bool

Sends the query to the instrument and returns the response as boolean.

query_int(*query: str*) → int

Sends the query to the instrument and returns the response as integer.

query_float(*query: str*) → float

Sends the query to the instrument and returns the response as float.

write_str_with_opc(*cmd: str, timeout: int = None*) → None

Writes the opc-synced command to the instrument. If you do not provide timeout, the method uses current `opc_timeout`.

write_with_opc(*cmd: str, timeout: int = None*) → None

This method is an alias to the `write_str_with_opc()`. Writes the opc-synced command to the instrument. If you do not provide timeout, the method uses current `opc_timeout`.

query_str_with_opc(*query: str, timeout: int = None*) → str

Sends the opc-synced query to the instrument and returns the response as string. The response is trimmed of any trailing LF characters and has no length limit. If you do not provide timeout, the method uses current `opc_timeout`.

query_with_opc(*query: str, timeout: int = None*) → str

This method is an alias to the `query_str_with_opc()`. Sends the opc-synced query to the instrument and returns the response as string. The response is trimmed of any trailing LF characters and has no length limit. If you do not provide timeout, the method uses current `opc_timeout`.

query_bool_with_opc(*query: str, timeout: int = None*) → bool

Sends the opc-synced query to the instrument and returns the response as boolean. If you do not provide timeout, the method uses current `opc_timeout`.

query_int_with_opc(*query: str, timeout: int = None*) → int

Sends the opc-synced query to the instrument and returns the response as integer. If you do not provide timeout, the method uses current `opc_timeout`.

query_float_with_opc(*query: str, timeout: int = None*) → float

Sends the opc-synced query to the instrument and returns the response as float. If you do not provide timeout, the method uses current `opc_timeout`.

write_bin_block(*cmd: str, payload: bytes*) → None

Writes all the payload as binary data block to the instrument. The binary data header is added at the beginning of the transmission automatically, do not include it in the payload!!!

query_bin_block(*query: str*) → bytes

Queries binary data block to bytes. Throws an exception if the returned data was not a binary data. Returns `data:bytes`

query_bin_block_with_opc(*query: str, timeout: int = None*) → bytes

Sends a OPC-synced query and returns binary data block to bytes. If you do not provide timeout, the method uses current `opc_timeout`.

query_bin_or_ascii_float_list(*query: str*) → List[float]

Queries a list of floating-point numbers that can be returned in ASCII format or in binary format. - For ASCII format, the list numbers are decoded as comma-separated values. - For Binary Format, the numbers are decoded based on the property `BinFloatFormat`, usually float 32-bit (FORM REAL,32).

query_bin_or_ascii_float_list_with_opc(*query: str, timeout: int = None*) → List[float]

Sends a OPC-synced query and reads a list of floating-point numbers that can be returned in ASCII format or in binary format. - For ASCII format, the list numbers are decoded as comma-separated values. - For Binary Format, the numbers are decoded based on the property `BinFloatFormat`, usually float 32-bit (FORM REAL,32). If you do not provide timeout, the method uses current `opc_timeout`.

query_bin_or_ascii_int_list(*query: str*) → List[int]

Queries a list of floating-point numbers that can be returned in ASCII format or in binary format. - For ASCII format, the list numbers are decoded as comma-separated values. - For Binary Format, the numbers are decoded based on the property BinFloatFormat, usually float 32-bit (FORM REAL,32).

query_bin_or_ascii_int_list_with_opc(*query: str, timeout: int = None*) → List[int]

Sends a OPC-synced query and reads a list of floating-point numbers that can be returned in ASCII format or in binary format. - For ASCII format, the list numbers are decoded as comma-separated values. - For Binary Format, the numbers are decoded based on the property BinFloatFormat, usually float 32-bit (FORM REAL,32). If you do not provide timeout, the method uses current `opc_timeout`.

query_bin_block_to_file(*query: str, file_path: str, append: bool = False*) → None

Queries binary data block to the provided file. If `append` is `False`, any existing file content is discarded. If `append` is `True`, the new content is added to the end of the existing file, or if the file does not exist, it is created. Throws an exception if the returned data was not a binary data. Example for transferring a file from Instrument -> PC: `query = f"MMEM:DATA? '{INSTR_FILE_PATH}'"`. Alternatively, use the dedicated methods for this purpose:

- `send_file_from_pc_to_instrument()`
- `read_file_from_instrument_to_pc()`

query_bin_block_to_file_with_opc(*query: str, file_path: str, append: bool = False, timeout: int = None*) → None

Sends a OPC-synced query and writes the returned data to the provided file. If `append` is `False`, any existing file content is discarded. If `append` is `True`, the new content is added to the end of the existing file, or if the file does not exist, it is created. Throws an exception if the returned data was not a binary data.

write_bin_block_from_file(*cmd: str, file_path: str*) → None

Writes data from the file as binary data block to the instrument using the provided command. Example for transferring a file from PC -> Instrument: `cmd = f"MMEM:DATA '{INSTR_FILE_PATH}'"`. Alternatively, use the dedicated methods for this purpose:

- `send_file_from_pc_to_instrument()`
- `read_file_from_instrument_to_pc()`

send_file_from_pc_to_instrument(*source_pc_file: str, target_instr_file: str*) → None

SCPI Command: `MMEM:DATA`

Sends file from PC to the instrument

read_file_from_instrument_to_pc(*source_instr_file: str, target_pc_file: str, append_to_pc_file: bool = False*) → None

SCPI Command: `MMEM:DATA?`

Reads file from instrument to the PC.

Set the `append_to_pc_file` to `True` if you want to append the read content to the end of the existing PC file

get_last_sent_cmd() → str

Returns the last commands sent to the instrument. Only works in simulation mode

go_to_local() → None

Puts the instrument into local state.

go_to_remote() → None

Puts the instrument into remote state.

get_lock() → RLock

Returns the thread lock for the current session.

By default:

- If you create standard new RsCMPX_WcdmaMeas instance with new VISA session, the session gets a new thread lock. You can assign it to other RsCMPX_WcdmaMeas sessions in order to share one physical instrument with a multi-thread access.
- If you create new RsCMPX_WcdmaMeas from an existing session, the thread lock is shared automatically making both instances multi-thread safe.

You can always assign new thread lock by calling `driver.utilities.assign_lock()`

assign_lock(lock: RLock) → None

Assigns the provided thread lock.

clear_lock()

Clears the existing thread lock, making the current session thread-independent from others that might share the current thread lock.

sync_from(source: Utilities) → None

Synchronises these Utils with the source.

RSCMPX_WCDMAMEAS LOGGER

Check the usage in the Getting Started chapter [here](#).

class ScpiLogger

Base class for SCPI logging

mode

Sets the logging ON or OFF. Additionally, you can set the logging ON only for errors. Possible values:

- `LoggingMode.Off` - logging is switched OFF
- `LoggingMode.On` - logging is switched ON
- `LoggingMode.Errors` - logging is switched ON, but only for error entries
- `LoggingMode.Default` - sets the logging to default - the value you have set with `logger.default_mode`

default_mode

Sets / returns the default logging mode. You can recall the default mode by calling the `logger.mode = LoggingMode.Default`.

Data Type

`LoggingMode`

device_name: str

Use this property to change the resource name in the log from the default Resource Name (e.g. `TCPIP::192.168.2.101::INSTR`) to another name e.g. `'MySigGen1'`.

set_logging_target(target, console_log: bool = None, udp_log: bool = None) → None

Sets logging target - the target must implement `write()` and `flush()`. You can optionally set the console and UDP logging ON or OFF. This method switches the logging target global OFF.

get_logging_target()

Based on the `global_mode`, it returns the logging target: either the local or the global one.

set_logging_target_global(console_log: bool = None, udp_log: bool = None) → None

Sets logging target to global. The global target must be defined. You can optionally set the console and UDP logging ON or OFF.

log_to_console

Returns logging to console status.

log_to_udp

Returns logging to UDP status.

log_to_console_and_udp

Returns true, if both logging to UDP and console in are True.

info_raw(log_entry: str, add_new_line: bool = True) → None

Method for logging the raw string without any formatting.

info(start_time: datetime, end_time: datetime, log_string_info: str, log_string: str) → None

Method for logging one info entry. For binary log_string, use the info_bin()

error(start_time: datetime, end_time: datetime, log_string_info: str, log_string: str) → None

Method for logging one error entry.

set_relative_timestamp(timestamp: datetime) → None

If set, the further timestamps will be relative to the entered time.

set_relative_timestamp_now() → None

Sets the relative timestamp to the current time.

get_relative_timestamp() → datetime

Based on the global_mode, it returns the relative timestamp: either the local or the global one.

clear_relative_timestamp() → None

Clears the reference time, and the further logging continues with absolute times.

flush() → None

Flush all the entries.

log_status_check_ok

Sets / returns the current status of status checking OK. If True (default), the log contains logging of the status checking 'Status check: OK'. If False, the 'Status check: OK' is skipped - the log is more compact. Errors will still be logged.

clear_cached_entries() → None

Clears potential cached log entries. Cached log entries are generated when the Logging is ON, but no target has been defined yet.

set_format_string(value: str, line_divider: str = '\n') → None

Sets new format string and line divider. If you just want to set the line divider, set the format string value=None. The original format string is: PAD_LEFT12(%START_TIME%) PAD_LEFT25(%DEVICE_NAME%) PAD_LEFT12(%DURATION%) %LOG_STRING_INFO% %LOG_STRING%

restore_format_string() → None

Restores the original format string and the line divider to LF

abbreviated_max_len_ascii: int

Defines the maximum length of one ASCII log entry. Default value is 200 characters.

abbreviated_max_len_bin: int

Defines the maximum length of one Binary log entry. Default value is 2048 bytes.

abbreviated_max_len_list: int

Defines the maximum length of one list entry. Default value is 100 elements.

bin_line_block_size: int

Defines number of bytes to display in one line. Default value is 16 bytes.

udp_port

Returns udp logging port.

target_auto_flushing

Returns status of the auto-flushing for the logging target.

RSCMPX_WCDMAMEAS EVENTS

Check the usage in the Getting Started chapter [here](#).

class Events

Common Events class. Event-related methods and properties. Here you can set all the event handlers.

property before_query_handler: Callable

Returns the handler of before_query events.

Returns

current before_query_handler

property before_write_handler: Callable

Returns the handler of before_write events.

Returns

current before_write_handler

property io_events_include_data: bool

Returns the current state of the io_events_include_data See the setter for more details.

property on_read_handler: Callable

Returns the handler of on_read events.

Returns

current on_read_handler

property on_write_handler: Callable

Returns the handler of on_write events.

Returns

current on_write_handler

sync_from(source: Events) → None

Synchronises these Events with the source.

**CHAPTER
TEN**

INDEX

INDEX

A

abbreviated_max_len_ascii (*ScpiLogger attribute*), 582
 abbreviated_max_len_bin (*ScpiLogger attribute*), 582
 abbreviated_max_len_list (*ScpiLogger attribute*), 582
 ABORT:WCDma:MEASurement<instance>:MEvaluation, 184
 ABORT:WCDma:MEASurement<instance>:OLPControl, 512
 ABORT:WCDma:MEASurement<instance>:OOSync, 518
 ABORT:WCDma:MEASurement<instance>:PRACH, 521
 ABORT:WCDma:MEASurement<instance>:TPC, 542

B

bin_line_block_size (*ScpiLogger attribute*), 582

C

CALCulate:WCDma:MEASurement<instance>:MEvaluation:SPECtrum, 471
 CALCulate:WCDma:MEASurement<instance>:MEvaluation:SPECtrum, 474
 CALCulate:WCDma:MEASurement<instance>:OLPControl, 512
 CALCulate:WCDma:MEASurement<instance>:OOSync, 518
 CALCulate:WCDma:MEASurement<instance>:PRACH:OFFPower, 523
 CALCulate:WCDma:MEASurement<instance>:PRACH:PREamble<nr>:C, 524
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>, 544
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>, 547
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>, 550
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>, 554
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>:MODulation:AVERAge, 556
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>:MODulation:CURRENT, 558
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>:MODulation:MAXimum, 559
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>:SDEVIation, 561
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>:RCDerror:AVERAge, 564
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>:RCDerror:CURRENT, 565
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>:RCDerror:MAXimum, 567
 CALCulate:WCDma:MEASurement<instance>:TPC:CARRIER<carrier>:RCDerror:SDEVIation, 567
 clear_cached_entries() (*ScpiLogger method*), 582
 clear_relative_timestamp() (*ScpiLogger method*), 582
 CALCulate:WCDma:MEASurement<instance>:MEvaluation:MODulation:PHDHSdpccch, 463
 CALCulate:WCDma:MEASurement<instance>:MEvaluation:MODulation:UEPHd, 464
 CALCulate:WCDma:MEASurement<instance>:MEvaluation:SPECtrum:AVERAge, 46
 CONFIGure:WCDma:MEASurement<instance>:CARRIER<carrier>:BAN, 45
 CONFIGure:WCDma:MEASurement<instance>:CELL:CARRIER<carrier>, 46
 CONFIGure:WCDma:MEASurement<instance>:MEvaluation:AMode:MO, 46

102	109
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:OOSync:TOUT,	
92	108
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:CFError,	
92	112
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:EVMagnit	
92	114
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:IQImbala	
92	112
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:IQOffset	
92	112
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:MERRor,	
92	115
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:PCONTrol	
92	116
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:PCONTrol	
92	116
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:PCONTrol	
92	118
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:PCONTrol	
92	116
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:PCONTrol	
92	115
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:PCONTrol	
92	119
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:PCONTrol	
104	120
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:LIMit:PERror,	
47	120
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:MOEXception,	
104	110
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:MPReamble,	
104	110
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:OFFPower,	
104	110
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:PPReamble,	
106	110
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:RESult:CHIP:EV	
47	126
CONFIGure:WCDma:MEASurement<instance>:MEValueatCONF:RESult:WCDma:MEASurement<instance>:PRACH:RESult:CHIP:ME	
47	126
CONFIGure:WCDma:MEASurement<instance>:OLPContrONF:RESult:WCDma:MEASurement<instance>:PRACH:RESult:CHIP:PE	
106	126
CONFIGure:WCDma:MEASurement<instance>:OLPContrONF:RESult:WCDma:MEASurement<instance>:PRACH:RESult:CHIP:UE	
106	126
CONFIGure:WCDma:MEASurement<instance>:OLPContrONF:RESult:WCDma:MEASurement<instance>:PRACH:RESult:EVMagni	
106	121
CONFIGure:WCDma:MEASurement<instance>:OOSync:ADDPChieveWCDma:MEASurement<instance>:PRACH:RESult:FERRor,	
108	121
CONFIGure:WCDma:MEASurement<instance>:OOSync:CONF:OFFUpdaWCDma:MEASurement<instance>:PRACH:RESult:IQ,	
109	121
CONFIGure:WCDma:MEASurement<instance>:OOSync:CONF:ONUpdaWCDma:MEASurement<instance>:PRACH:RESult:MERRor,	
109	121
CONFIGure:WCDma:MEASurement<instance>:OOSync:CONF:RESult:WCDma:MEASurement<instance>:PRACH:RESult:PERror,	

121	145
CONFIGure:WCDma:MEASurement<instance>:PRACH:RESOnse:PSSteps	CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:ILPControl
121	146
CONFIGure:WCDma:MEASurement<instance>:PRACH:RESOnse:UEPower	CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:ILPControl
121	147
CONFIGure:WCDma:MEASurement<instance>:PRACH:RESOnse:ULPower	CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:MPEDch,
121	148
CONFIGure:WCDma:MEASurement<instance>:PRACH:TOUT	CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:ULCM:PA,
110	149
CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:LIMit:ULCM:PB,	CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:LIMit:ULCM:PB,
130	150
CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:MODE,	CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:MODE,
131	131
CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:MOEXception,	CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:MOEXception,
128	131
CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:MONitor:MLENght,	CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:MONitor:MLENght,
128	151
CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:MPEDch:AEXecutio	CONFIGure:WCDma:MEASurement<instance>:RFSettings:CONFiguration:WCDma:MEASurement<instance>:TPC:MPEDch:AEXecutio
128	151
CONFIGure:WCDma:MEASurement<instance>:TPC:CSEL	CONFIGure:WCDma:MEASurement<instance>:TPC:MPEDch:MLENght,
131	151
CONFIGure:WCDma:MEASurement<instance>:TPC:CTFCONENgth	CONFIGure:WCDma:MEASurement<instance>:TPC:SETup,
134	131
CONFIGure:WCDma:MEASurement<instance>:TPC:DHIBONENgth	CONFIGure:WCDma:MEASurement<instance>:TPC:TOUT,
135	131
CONFIGure:WCDma:MEASurement<instance>:TPC:DHIBONENgth	CONFIGure:WCDma:MEASurement<instance>:TPC:ULCM:AEXecution,
135	152
CONFIGure:WCDma:MEASurement<instance>:TPC:DHIBONENgth	CONFIGure:WCDma:MEASurement<instance>:TPC:ULCM:MLENght,
135	152
CONFIGure:WCDma:MEASurement<instance>:TPC:ILPControl:AEXecution	CONFIGure:WCDma:MEASurement<instance>:UEChannels:BSFSelect
136	153
CONFIGure:WCDma:MEASurement<instance>:TPC:ILPControl:WCDma:MEASurement<instance>:UEChannels:CARRier<	CONFIGure:WCDma:MEASurement<instance>:UEChannels:CARRier<
136	154
CONFIGure:WCDma:MEASurement<instance>:TPC:ILPControl:WCDma:MEASurement<instance>:UEChannels:CARRier<	CONFIGure:WCDma:MEASurement<instance>:UEChannels:CARRier<
137	155
CONFIGure:WCDma:MEASurement<instance>:TPC:ILPControl:WCDma:MEASurement<instance>:UEChannels:CARRier<	CONFIGure:WCDma:MEASurement<instance>:UEChannels:CARRier<
138	156
CONFIGure:WCDma:MEASurement<instance>:TPC:ILPControl:WCDma:MEASurement<instance>:UEChannels:CARRier<	CONFIGure:WCDma:MEASurement<instance>:UEChannels:CARRier<
136	157
CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:CONFiguration:WCDma:MEASurement<instance>:UEChannels:CARRier<	CONFIGure:WCDma:MEASurement<instance>:UEChannels:CARRier<
140	159
CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:CONFiguration:WCDma:MEASurement<instance>:UEChannels:CARRier<	CONFIGure:WCDma:MEASurement<instance>:UEChannels:CARRier<
139	160
CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:CONFiguration:WCDma:MEASurement<instance>:UESignal:CARRier<car	CONFIGure:WCDma:MEASurement<instance>:UESignal:CARRier<car
141	163
CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:CONFiguration:WCDma:MEASurement<instance>:UESignal:CMPattern,	CONFIGure:WCDma:MEASurement<instance>:UESignal:CMPattern,
142	161
CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:CONFiguration:WCDma:MEASurement<instance>:UESignal:DPDCh,	CONFIGure:WCDma:MEASurement<instance>:UESignal:DPDCh,
142	161
CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:CONFiguration:WCDma:MEASurement<instance>:UESignal:SFORMAT,	CONFIGure:WCDma:MEASurement<instance>:UESignal:SFORMAT,
144	161
CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:CONFiguration:WCDma:MEASurement<instance>:UESignal:ULConfig,	CONFIGure:WCDma:MEASurement<instance>:UESignal:ULConfig,
142	161
CONFIGure:WCDma:MEASurement<instance>:TPC:LIMit:ILPControl:MINPower,	

D

default_mode (*ScpiLogger attribute*), 581device_name (*ScpiLogger attribute*), 581

E

error() (*ScpiLogger method*), 582

F

FETCh:WCDMa:MEASurement<instance>:MEValuation:BER,

186

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:CDERror:AVERage,

187

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:CDERror:CURrent,

188

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:CDERror:MAXimum,

190

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:CDERror:SDEVIation,

191

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:CDPower:AVERage,

192

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:CDPower:CURrent,

193

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:CDPower:MAXimum,

194

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:CDPower:MINimum,

195

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:CDPower:SDEVIation,

197

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:MODulation:AVERage,

198

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:MODulation:CURrent,

200

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:MODulation:MAXimum,

202

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:MODulation:SDEVIation,

204

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:RCDerror:AVERage,

207

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:RCDerror:CURrent,

209

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:RCDerror:MAXimum,

211

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:RCDerror:OcInfo,

213

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:RCDerror:SDEVIation,

215

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:RCDerror:SF,

214

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:TRACE:CDERror:DPCCh:AVERage,

218

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:TRACE:CDERror:DPCCh:CURrent,

219

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>:TRACE:CDERror:DPCCh:MAXimum,

220

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

221

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

222

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

223

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

224

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

225

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

227

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

228

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

229

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

230

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

232

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

233

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

234

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

235

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

237

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

238

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

239

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

240

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

242

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

243

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

244

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

245

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

246

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

247

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

248

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

249

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

250

FETCh:WCDMa:MEASurement<instance>:MEValuation:CARRIER<carrier>

251

Index 593

[illegible]

FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:RMS:Current:MEvaluation:LIST:SEGMENT
 393 409
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:PEAK:AVERAGE:MEvaluation:LIST:SEGMENT
 394 410
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:PEAK:CURRENT:MEvaluation:LIST:SEGMENT
 394 411
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:PEAK:MAXIMUM:MEvaluation:LIST:SEGMENT
 395 412
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:PEAK:SD:MEvaluation:LIST:SEGMENT
 395 413
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:RMS:AVERAGE:MEvaluation:LIST:SEGMENT
 396 414
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:RMS:CURRENT:MEvaluation:LIST:SEGMENT
 396 415
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:RMS:MAXIMUM:MEvaluation:LIST:SEGMENT
 397 416
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:RMS:SD:MEvaluation:LIST:SEGMENT
 397 417
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:SD:MEvaluation:LIST:SEGMENT
 398 418
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:AVSRANGE:MEvaluation:LIST:SEGMENT
 399 419
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:CURRANGE:MEvaluation:LIST:SEGMENT
 399 420
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:MAXIMUM:MEvaluation:LIST:SEGMENT
 400 421
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:SD:MEvaluation:LIST:SEGMENT
 400 422
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:AVSRANGE:MEvaluation:LIST:SEGMENT
 401 423
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:CURRANGE:MEvaluation:LIST:SEGMENT
 401 424
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:MAXIMUM:MEvaluation:LIST:SEGMENT
 402 425
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:SD:MEvaluation:LIST:SEGMENT
 402 426
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:AVSRANGE:MEvaluation:LIST:SEGMENT
 403 427
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:CURRANGE:MEvaluation:LIST:SEGMENT
 404 429
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:MAXIMUM:MEvaluation:LIST:SEGMENT
 404 431
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:SD:MEvaluation:LIST:SEGMENT
 405 431
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:AVSRANGE:MEvaluation:LIST:SEGMENT
 405 432
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:CURRANGE:MEvaluation:LIST:SEGMENT
 406 433
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:MAXIMUM:MEvaluation:LIST:SEGMENT
 407 434
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:SD:MEvaluation:LIST:SEGMENT
 407 434
 FETCH:WCDMA:MEASurement<instance>:MEvaluation:FHSC:WCDMA:MEASUREMENT:AVSRANGE:MEvaluation:LIST:SEGMENT
 408 435


```

483 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:WCDMa:MEASurement<instance>:CURRENT;MEvaluation:TRACE:PHD:CURRENT;
484 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:WCDMa:MEASurement<instance>:CURRENT;OLPControl,
485 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:HKFRight:AVERAGE;OLPControl:CARRIER<carrier>,
486 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:HKFRight:CURRENT;OLPControl:STATE,
487 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:HKFRight:MAXimum;OLPControl:STATE:ALL,
489 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:HKFRight:AVERAGE;OOSync, 518
490 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:HKFRight:CURRENT,
491 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:HKFRight:MAXimum,
492 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:KFILter:AVERAGE,
493 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:KFILter:CURRENT,
494 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:KFILter:MAXimum,
495 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:MFLeft:AVERAGE,
496 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:MFLeft:CURRENT,
497 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:MFLeft:MAXimum,
499 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:MFRight:AVERAGE,
500 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:MFRight:CURRENT,
501 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EMASK:MFRight:MAXimum,
502 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EVMagnitude:CHIP:AVERAGE,
503 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EVMagnitude:CHIP:CURRENT,
504 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:EVMagnitude:CHIP:MAXimum,
505 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:IQ30CURRENT,
506 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:MERRor:CHIP:AVERAGE,
507 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:MERRor:CHIP:CURRENT,
508 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:MERRor:CHIP:MAXimum,
509 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:PERRor:CHIP:AVERAGE,
510 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:PERRor:CHIP:CURRENT,
510 FETCH:WCDMa:MEASurement<instance>:MEvaluation:TRACE:PERRor:CHIP:MAXimum,

```

547 INITiate:WCDMa:MEASurement<instance>:OOSync,
 FETCH:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:MINimum,
 550 INITiate:WCDMa:MEASurement<instance>:PRACH,
 FETCH:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:STATISTICS,
 552 INITiate:WCDMa:MEASurement<instance>:TPC: 542
 FETCH:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:TRACE:PSTeps:CURRENT,
 554
 FETCH:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:TRACE:UEPower:CURRENT,
 556 log_status_check (ScpiLogger attribute), 582
 log_to_console (ScpiLogger attribute), 581
 FETCH:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:AVERAGE,
 558 log_to_console_and_udp (ScpiLogger attribute), 581
 log_to_udp (ScpiLogger attribute), 581
 FETCH:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:MAXimum,
 559
 FETCH:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:MINimum,
 561 mode (ScpiLogger attribute), 581
 FETCH:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:STATISTICS,
 563
 FETCH:WCDMa:MEASurement<instance>:TPC:DHIB:AVERAGE, 186
 564
 FETCH:WCDMa:MEASurement<instance>:TPC:DHIB:MAXimum, 187
 565
 FETCH:WCDMa:MEASurement<instance>:TPC:DHIB:MINimum, 188
 566
 FETCH:WCDMa:MEASurement<instance>:TPC:DHIB:STATISTICS, 190
 568
 FETCH:WCDMa:MEASurement<instance>:TPC:STATE, 191
 568
 FETCH:WCDMa:MEASurement<instance>:TPC:STATE:ALL, 192
 569
 FETCH:WCDMa:MEASurement<instance>:TPC:TOTAL:TRACE:UEPower:CURRENT,
 571 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:MINimum,
 572 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:STATISTICS,
 572 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:CURRENT,
 572 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:AVERAGE,
 572 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:MAXimum,
 572 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:MINimum,
 573 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:STATISTICS,
 574
 flush() (ScpiLogger method), 582
 200
 G READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:MINimum,
 202
 get_logging_target() (ScpiLogger method), 581
 get_relative_timestamp() (ScpiLogger method), 204
 582
 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:MINimum,
 207
 I READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:MINimum,
 209
 info() (ScpiLogger method), 582
 info_raw() (ScpiLogger method), 581
 INITiate:WCDMa:MEASurement<instance>:MEvaluation, 211
 184
 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:MINimum,
 213
 INITiate:WCDMa:MEASurement<instance>:OLPControl, 213
 512
 READ:WCDMa:MEASurement<instance>:MEvaluation:CARRIER<carrier>:PSTeps:MINimum,
 215

Index	599
--------------	------------

READ:WCDMa:MEASurement<instance>:MEValuation:MODULatION:PHASSubpenh<instance>:MEValuation:TRACe:EVMagnit
 463 502
 READ:WCDMa:MEASurement<instance>:MEValuation:MODULatION:WEPASurement<instance>:MEValuation:TRACe:EVMagnit
 464 503
 READ:WCDMa:MEASurement<instance>:MEValuation:PRACH:CURRENTMEASurement<instance>:MEValuation:TRACe:EVMagnit
 466 504
 READ:WCDMa:MEASurement<instance>:MEValuation:PRACH:MAXimumMEASurement<instance>:MEValuation:TRACe:IQ:CURR
 467 505
 READ:WCDMa:MEASurement<instance>:MEValuation:SPEDr:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:
 468 506
 READ:WCDMa:MEASurement<instance>:MEValuation:SPEDr:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:
 471 507
 READ:WCDMa:MEASurement<instance>:MEValuation:SPEDr:WCDMa:MEASurement<instance>:MEValuation:TRACe:MERRor:
 474 508
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:CARRier:MEASurement:PERRor[<RMS>]EVALuent:TRACe:PERRor:
 479 509
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:WCDMa:MEASurement:ISigmalin:CURRent>:MEValuation:TRACe:PERRor:
 480 510
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:WCDMa:MEASurement:QSigmalin:CURRent>:MEValuation:TRACe:PERRor:
 482 510
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:WCDMa:MEASurement:ISigmalin:CURRent>:MEValuation:TRACe:PHD:CUR
 483 511
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:WCDMa:MEASurement:QSigmalin:CURRent>:OLPControl,
 484 512
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:WCDMa:MEASurement:WEPASurement<instance>:OLPControl:CARRier<carrie
 485 515
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:WCDMa:MEASurement:CURRentistance>:OOSync, 518
 486 READ:WCDMa:MEASurement<instance>:PRACH:OFFPower,
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:HKFLeft:MAXimum,
 487 READ:WCDMa:MEASurement<instance>:PRACH:PREAmble<nr>:CURRent
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:HKFRight:AVERAge,
 489 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude:CUR
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:HKFRight:CURRent,
 490 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude:CUR
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:HKFRight:MAXimum,
 491 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:EVMagnitude[:
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:KFILter:AVERAge,
 492 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:FERRor:CURRent
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:KFILter:CURRent,
 493 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:IQ:CURRent,
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:KFILter:MAXimum,
 494 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:MERRor:CHIP:CUR
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:MFLeft:AVERAge,
 495 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:MERRor:PEAK:CUR
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:MFLeft:CURRent,
 496 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:MERRor[:RMS]:CUR
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:MFLeft:MAXimum,
 497 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:PERRor:CHIP:CUR
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:MFRight:AVERAge,
 499 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:PERRor:PEAK:CUR
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:MFRight:CURRent,
 500 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:PERRor[:RMS]:CUR
 READ:WCDMa:MEASurement<instance>:MEValuation:TRACe:EMASK:MFRight:MAXimum,
 501 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:PSTeps:CURRent

539 S
 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:UEPower:CHIP:CURRENT, ScpiLogger (class in RsCMPX_WcdmaMeas.Internal.ScpiLogger),
 540 581
 READ:WCDMa:MEASurement<instance>:PRACH:TRACe:UEPower:CURRENT, set_format_string() (ScpiLogger method), 582
 541 set_logging_target() (ScpiLogger method), 581
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:AVERAGE, set_logging_target_global() (ScpiLogger method),
 544 581
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:MAXimum, set_relative_timestamp() (ScpiLogger method),
 547 582
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:MINimum, set_relative_timestamp_now() (ScpiLogger
 550 method), 582
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:PSTeps:STATISTICS, STOP:WCDMa:MEASurement<instance>:MEvaluation,
 552 184
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:TRACe:PSTeps:CURRENT, STOP:WCDMa:MEASurement<instance>:OLPControl,
 554 512
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:TRACe:UEPower:CURRENT, STOP:WCDMa:MEASurement<instance>:OOSync, 518
 556 STOP:WCDMa:MEASurement<instance>:PRACH, 521
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:AVERAGE, STOP:WCDMa:MEASurement<instance>:TPC, 542
 558
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:MAXimum, target_auto_flushing (ScpiLogger attribute), 582
 559
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:MINimum, TRIGGER:WCDMa:MEASurement<instance>:MEvaluation:CATalog:SC
 561 171
 READ:WCDMa:MEASurement<instance>:TPC:CARRIER<carrier>:UEPower:STATISTICS, TRIGGER:WCDMa:MEASurement<instance>:MEvaluation:DElay,
 563 167
 READ:WCDMa:MEASurement<instance>:TPC:DHIB:AVERAGE, TRIGGER:WCDMa:MEASurement<instance>:MEvaluation:LIST:MODE,
 564 171
 READ:WCDMa:MEASurement<instance>:TPC:DHIB:MAXimum, TRIGGER:WCDMa:MEASurement<instance>:MEvaluation:MGAP,
 565 167
 READ:WCDMa:MEASurement<instance>:TPC:DHIB:MINimum, TRIGGER:WCDMa:MEASurement<instance>:MEvaluation:SLOPe,
 566 167
 READ:WCDMa:MEASurement<instance>:TPC:DHIB:STATISTICS, TRIGGER:WCDMa:MEASurement<instance>:MEvaluation:SOURce,
 568 167
 READ:WCDMa:MEASurement<instance>:TPC:TOTal:TRACe:UEPower:CURRENT, TRIGGER:WCDMa:MEASurement<instance>:MEvaluation:THReshold,
 571 167
 READ:WCDMa:MEASurement<instance>:TPC:TOTal:UEPower:AVERAGE, TRIGGER:WCDMa:MEASurement<instance>:MEvaluation:TOUT,
 572 167
 READ:WCDMa:MEASurement<instance>:TPC:TOTal:UEPower:MAXimum, TRIGGER:WCDMa:MEASurement<instance>:OLPControl:CATalog:SO
 572 174
 READ:WCDMa:MEASurement<instance>:TPC:TOTal:UEPower:MINimum, TRIGGER:WCDMa:MEASurement<instance>:OLPControl:DElay,
 573 172
 READ:WCDMa:MEASurement<instance>:TPC:TOTal:UEPower:STATISTICS, TRIGGER:WCDMa:MEASurement<instance>:OLPControl:MGAP,
 574 172
 restore_format_string() (ScpiLogger method), 582
 ROUTe:WCDMa:MEASurement<instance>, 164
 ROUTe:WCDMa:MEASurement<instance>:SCENario, TRIGGER:WCDMa:MEASurement<instance>:OLPControl:SOURce,
 165 172
 ROUTe:WCDMa:MEASurement<instance>:SCENario:CSPath, TRIGGER:WCDMa:MEASurement<instance>:OLPControl:THReshold,
 165 172
 ROUTe:WCDMa:MEASurement<instance>:SCENario:MAProtocol, TRIGGER:WCDMa:MEASurement<instance>:OLPControl:TOUT,
 166 172
 ROUTe:WCDMa:MEASurement<instance>:SCENario:SALone, TRIGGER:WCDMa:MEASurement<instance>:OOSync:CATalog:SOURce,
 166 177

TRIGger:WCDMa:MEASurement<instance>:OOSync:DElay,
 175
 TRIGger:WCDMa:MEASurement<instance>:OOSync:MGAP,
 175
 TRIGger:WCDMa:MEASurement<instance>:OOSync:SLOPe,
 175
 TRIGger:WCDMa:MEASurement<instance>:OOSync:SOURce,
 175
 TRIGger:WCDMa:MEASurement<instance>:OOSync:THReshold,
 175
 TRIGger:WCDMa:MEASurement<instance>:OOSync:TOUT,
 175
 TRIGger:WCDMa:MEASurement<instance>:PRACH:CATalog:SOURce,
 180
 TRIGger:WCDMa:MEASurement<instance>:PRACH:DElay,
 178
 TRIGger:WCDMa:MEASurement<instance>:PRACH:MGAP,
 178
 TRIGger:WCDMa:MEASurement<instance>:PRACH:SLOPe,
 178
 TRIGger:WCDMa:MEASurement<instance>:PRACH:SOURce,
 178
 TRIGger:WCDMa:MEASurement<instance>:PRACH:THReshold,
 178
 TRIGger:WCDMa:MEASurement<instance>:PRACH:TOUT,
 178
 TRIGger:WCDMa:MEASurement<instance>:TPC:CATalog:SOURce,
 184
 TRIGger:WCDMa:MEASurement<instance>:TPC:DElay,
 181
 TRIGger:WCDMa:MEASurement<instance>:TPC:MGAP,
 181
 TRIGger:WCDMa:MEASurement<instance>:TPC:SLOPe,
 181
 TRIGger:WCDMa:MEASurement<instance>:TPC:SOURce,
 181
 TRIGger:WCDMa:MEASurement<instance>:TPC:THReshold,
 181
 TRIGger:WCDMa:MEASurement<instance>:TPC:TOUT,
 181

U

udp_port (*ScpiLogger attribute*), 582