will'tek

Willtek 3100

Mobile Fault Finder

SCPI reference guide
version 1.00

Notice Every effort was made to ensure that the information in this document was accurate at the time of printing. However, information is subject to change without notice, and Willtek reserves the right to provide an addendum to this document with information not available at the time this document was created.

Trademarks Willtek is a trademark of Willtek Communications GmbH in Germany and other countries.

Ordering information This guide is issued as part of the **3100 Mobile Fault Finder**. The ordering number for a published guide is M 293 111. The ordering number for the product is M 101 110.

Table of Contents

SCPI

1

This chapter describes the remote control capabilities of the 3100. The topics described in this chapter are as follows:

Overview

SCPI (pronounced as 'skippy') is a world-wide standard. The basic idea of SCPI is to define a command language for measurement systems that is independent of the related manufacturers.

This chapter explains how SCPI works on the 3100. It is divided into six subsections:

What SCPI is – Here you will find a short definition of SCPI and some general information.

Structure – This subsection explains the common commands, the subsystems and the command tree.

Syntax and notation – The command syntax, the parameters and the syntax of queries are explained in this subsection.

SCPI and RAPID! – SCPI and RAPID! together form a powerful time-saver for extensive tests that are performed on a regular basis, like automated acceptance tests after manufacturing or regular routine checks as part of maintenance or quality assurance.

Command subsystem overview – This subsection gives a brief overview on the subsystems into which the SCPI commands are split.

The Node B and UE parameter subsystems

The communication-related subsystems

SCPI command errors – Here, you will find a table with explanations of syntax errors, run-time errors, device errors or any other errors.

What SCPI is

SCPI (**S**tandard **C**ommands for **P**rogrammable **I**nstruments) was introduced in 1990. It is a world-wide standard, independent of single manufacturers.
The SCPI specifications define a command language for measurement systems and – in principle – are based on IEEE 488.2.
SCPI is independent of the physical transmission channel of the commands. Although this chapter refers to the GPIB as the physical interface, you can also control the 3100 via a local area network and USB.

The idea behind SCPI is to shorten program development times for the automated control of test equipment and to make that program development as efficient as possible.
Of course, one of the main requirements for this goal is that the language must be understood by as many measurement devices as possible. Therefore, SCPI is pushed by the SCPI consortium. Quite a number of the main test and measurement equipment manufacturers are members of the SCPI consortium.

Implementing just standard commands on a complex communications test system like the 3100 would lead to a poor performance. Therefore, we were obliged to find a compromise between standard commands and performance. This is the reason why you will find many more SCPI commands on the 3100 than specified in the standard SCPI specification.
However, all SCPI commands implemented on the 3100 follow the standard SCPI syntax and rules.

For additional details on the SCPI standard, the current version can be found on page www.scpiconsortium.org/scpistandard.htm. You can download the full SCPI specification from there free of charge (some 3.5 Mb in PDF format).

Structure

SCPI defines programming commands, program messages, return values and data formats, which are consistent for all measurement systems independent of their manufacturer and purpose.

SCPI uses a device-independent command set, the so-called Common commands, understood by all SCPI devices.
The 3100-specific SCPI commands are called Compound commands and will only be understood by the 3100 and its subsystems.
A subsystem in terms of SCPI is quite abstract: it is the set of commands implemented to perform specific tasks of the SCPI device (the 'measurement subsystem' of the 3100, for instance, is the set of commands implemented for taking all kind of measurements, while the 'configuration subsystem' is the set of all configuration commands for all areas of the 3100).
All these subsystems are using the same, SCPI-based messaging and data formats.

Any SCPI command is built in a hierarchical way – similar to how a path in a file system is built.
The single command elements are separated by colons (:).
The complete set of commands of a subsystem is called the 'command tree'.
The command trees for the SCPI command set implemented on the 3100 is available in this manual.

Syntax and notation

There are two types of SCPI commands:

- compound commands and
- common commands.

Both types of commands differ in syntax.

Compound commands

Compound commands are always referred to as **commands** throughout this chapter.

- Any compound command is built in a hierarchical way. The single command elements are separated by colons (:).
- A command usually starts with a colon (:). However, the colon **must be omitted** when the subsequent command continues on the same hierarchical level (see examples below).
- The first command element always is the name of the subsystem like `CONFigure` or `MEASure`.
- Then follows one of the commands available for that subsystem like `GSM` or `RFTX`.
- The subsequent command element(s) may now be one or more subcommand(s) and/or one or more parameter term(s) (e.g. `BS:ID:BCC 5`).
- A SCPI program line may contain more than one command. In this case, the single commands have to be separated by a semicolon (;).
- There is also a short form for every command. This is usually formed of the first four letters (`CONF` instead of `CONFIGURE`). When the fourth letter is a vowel (a,e,i,o,u), only the first three letters are used (`RFG` instead of `RFGenerator`).
- Throughout this manual, the short form is always written in capitals to make it easy to identify it. However, the SCPI system of the 3100 is not case-sensitive.

Examples

- The complete SCPI command to set the base station color code (i.e. the training sequence) to 5 would be: `:CONFigure:GSM:BS:ID:BCC 5`. The short form `:CONF:GSM:BS:ID:BCC 5` is completely identical.
- `:RFGenerator:STATe ON` is identical with `:RFG:STAT ON`. However, the long version of the commands is – especially in the beginning – easier to work with.
- `:RFGenerator:STATe ON; LEVel 20` is a valid two-command statement. `STATe ON` and `LEVel` are both commands of the `RFGenerator` subsystem. Both are working on the same hierarchical level. Therefore, the colon in front of `LEVel 20` has to be omitted.
  An alternative with the identical meaning would be: `:RFGenerator:STATe ON; :RFGenerator:LEVel 20`. The form `:RFGenerator:STATe ON; :LEVel 20` would be invalid, because `LEVel` is no subsystem of the 3100.
- `:RFGenerator:MODulation:STATe ON` is invalid. `STATe ON` is no subcommand of the `:RFG:MOD` level.
- `:RFGenerator:MODulation ON; FREQuency 850.2` In this case, the first part up to the semicolon(;) is valid. The second part is invalid as `FREQ` is no subcommand of the `:RFG:MOD` level.
  The correct command would be: `:RFGenerator:MODulation ON; :RFGenerator:FREQuency 850.2`.

Parameters

Many commands require parameters to be specified. Those parameters are placed behind the command, separated by at least one blank (space).
The following types of parameters do exist on the 3100:

- **Numeric parameters.** These are integers, floating point numbers (with a maximum of 6 decimals) or exponential numbers (see specifications IEEE 488.2 NRf format or ANSI X3.42-1975 for details).
  Some numeric parameters may carry a physical unit (details can be found in subsection variables of the RAPID! chapter).
  Example: `:RFG:FREQ 930.071965 MHz` makes the RF generator switch to the specified transmission frequency.

NOTE

The decimal point of floating point numbers **must be** the dot (.) within SCPI as a comma (,) will always be interpreted as the separator between two parameters (see Notes below for details).

- **Boolean parameters** are specified using the binary numbers 0 | 1, or ON | OFF instead.
- **Enumerated parameters** are strings that only may be selected from a predefined list. To distinguish them from string parameters, the enumerated parameters **must not** be put in quotation marks.
  Example: `:CONFigure:GSM:TYPE GSM9001800` sets the 3100 to work as test set for GSM 900/1800.
- **String parameters** are user-defined strings.
  Example: `PROGram:NAME "/rapid/examples/ms_test.rbm"` loads the specified RAPID! program file.
- **Block parameters** are a special case and will be described with the related commands.

Queries

Many commands also have a query form. These queries enable you to read out the current value of parameters or the results of measurements.
For a query, simply add a question mark to the command (without any spaces or other symbols in-between).
Example: `:RFG:FREQ?` returns the current transmission frequency of the RF generator.

NOTE

The result of a query is saved internally on the 3100. Details can be found in subsection "Using queries" on page 15.

Common commands

Common commands are defined in IEEE 488.2.
They work on the device itself (and on any subsystem) and always start with an asterisk (*).

Example: `*RST` resets the 3100 and sets all system parameters to default values.

A list of all common commands can be found in subsection "Common commands" on page 11.

NOTE

The SCPI system is not case-sensitive. It does not matter for the syntax whether commands are written using capital letters, lowercase letters, or a mixture of both.
However, for easy maintenance of SCPI programs, it is recommended to type in the short form of a command in capital letters (`CONF`) and the rest of it in lowercase letters (`CONFigure`).

NOTE

Some commands allow more than one parameter. In those cases, the single parameters are separated by commas (,). There **must not be** any spaces between the commas and the parameters.
Example: `:CONF:GSM:BS:TCH:NCELl 63,45,39,17,23,9`

NOTE

The SCPI notation of commands differs from the RAPID! notation. Please, do not confuse them as this could lead to severe program errors.

SCPI notation

The notation for SCPI commands is partly different from the RAPID! syntax. Please, do not confuse them as this could lead to severe program errors.

`[ item ]` (identical with RAPID!) — Square brackets indicate an optional item, which can also be omitted.
Example: `:MEASure[:CONTinuous]:RFTX:PPEak`. Regardless whether the `:CONTinuous` command element is used or not, the 3100 will start taking continuous measurements of the peak phase error.

`item | item` (identical with RAPID!) — Vertical bars separate entries of a list and indicate that precisely one element out of that list **must** be used.
Example: Some commands require boolean parameters to be specified. This means
that either on (1) or off (0) needs to follow the command. This is expressed by using vertical bars: `ON | OFF`.

`< item >` (does not exist with RAPID!) — Pointed brackets indicate that either a parameter or a subcommand must be used in order to build a valid command.
Example: `:RFGenerator:STATe <ON | OFF>`. The RF generator can be set either on or off.

`{ item }` (usage different to RAPID!) — Braces stand for a parameter or a subcommand that has to be selected from a predefined table.
Example: `:MEASure<:{measProp}>` indicates all subcommands of the `MEASure` subsystem, like `RFTX`, `RFRX` or `AFAN`.

SCPI and RAPID!

This section is an application-oriented guide on how to use the functionality of SCPI through RAPID!.

Executing SCPI commands through RAPID!

Reading SCPI data through RAPID!

Using queries

Event handling – registers

Programming examples

Executing SCPI commands

RAPID! treats the SCPI system like a file. This means that

- the SCPI system needs to be opened first, before any communication may be established.
- to communicate between RAPID! and SCPI, the standard file-related RAPID! commands (like PRINT or INPUT) are used.

Example 1 This example opens the SCPI system as a communication port:

```
LET scpi = FREEFILE
OPEN "scpi" as #scpi
...
...
...
CLOSE #scpi
```

Example 2 After the SCPI system has been opened as a communication port, commands may be sent, using the PRINT or OUTPUT command:

```
...
PRINT #scpi, "*RST"
OUTPUT #scpi, ":CONFigure:GSM:TYPE GSM9001800"
PRINT #scpi, ":MEASure:GSM:RFTX:ALL"
...
PRINT #scpi, ":FETCh:GSM:RFTX:ALL"
INPUT #scpi, rftx_result_all$ ...
```

The first command resets the 3100 to factory defaults using SCPI common command `*RST`.
The subsequent command configures the 3100 as a test set for GSM 900/1800.
The `:MEASure:RFTX:ALL` command starts an internal procedure that will continuously take all RFTX measurements in a row.
The current set of measurement results can be read out, using the `INPUT` command.

Example 3 Converting the string read into numeric variables.

```
...
index = 0
rftx_result_all$ = rftx_result_all$ + ","
DO

   P = INSTR(rftx_result_all$, ",")
   result(index) = VAL(LEFT$(rftx_result_all$, P -
   1))
   rftx_result_all$ = MID$(rftx_result_all$, P +1))
   index = index + 1

LOOP UNTIL rftx_result_all$=""
```

This example program looks for the commas, separating the single measurement result values. Then it reads the part of the string between the commas and converts it back into a numeric variable.
More details regarding this example program can be found in section "Standard TX measurements" on page 17.

Important notes:

- Please keep in mind that RAPID! has to check the GPIB actively as there will be no automated reaction to any GPIB control sequences. However, please note that measurement results, polled through GPIB commands will also be available for any RAPID! program.
- To make sure that the 3100 is not blocked by other tasks, only perform RAPID! or SCPI measurements from the Welcome menu.
- In order to prevent the 3100 from 'waiting for eternity', use a standard timeout of 10 s (allow more time for complicated measurements like RX testing).
- A timeout does not speed up things (or slow them down). A call establishment will take its time.

Reading SCPI data

To read SCPI data (such as measurement results) into RAPID!, the INPUT command is used.
Again, the SCPI system has to be opened as a communication port first.

NOTE

All results handed back from the SCPI system have the format of a string. Should there be more than one result, the single results will be separated by commas.

Example:

This example reads out the version of the 3100.

```
LET scpi = FREEFILE
OPEN "scpi" as #scpi
PRINT #scpi, "*IDN?"
INPUT #scpi, version$
CLOSE #scpi
```

SCPI common command `*IDN?` is used to read out the identification of the test set. The value returned is then stored in RAPID! string variable `version$`. The content of `version$` will be something like `"Willtek,3100,0511099"`.

Using queries

Many commands have a query form. These queries enable you to read out data from the SCPI system like measurement results or the current value of parameters (like the current transmission frequency of the RF generator).

Building queries

For a query, simply add a question mark to the command (without any spaces or other symbols in-between).

Example 1

The example below initiates the measurement of the burst length and returns the related value.

```
...
OUTPUT #scpi, ":MEASure:RFTX:LENGth?"
INPUT #scpi, burstlength$
```

Example 2

More than one query can be placed into a single command line. The individual queries must be separated by semicolons (;). The returned values will also be separated by semicolons.
The example below reads out the current system time and then queries the contents of the event status register.

```
...
OUTPUT #scpi, ":SYSTem:TIME?; :*ESR?"
INPUT #scpi, result$
```

The returned result string could be `"17,40,55;4"`, for example. This would denote a time of `5:40` PM and an event status register value of `4`.

Example 3

In case, a query needs a parameter to be specified, this parameter is then placed behind the question mark. As with normal commands, there must be at least one blank between the question mark and the parameter:

```
...
OUTPUT #scpi, ":MEASure:GSM:ARRay:RFRX:BER:CIA? 2"
...
```

This command will perform the BER measurement on the class Ia bits twice and return both measurement results in a string. The string returned in this example is `"2.1,2.2"`.

Event handling – registers

Basically, measurements can be started and results can be read out using the `MEASure` and `FETCh` subsystems of the 3100.
However, system events (like errors) could occur while those measurements are in progress and the measurement results read in could thus be invalid.
Therefore, the 3100 is equipped with an event handling system. This system may be programmed by the user to raise an event on certain conditions.
Using the programming features, the SCPI system of the 3100 will for instance generate an event in case an error occurs during the performance of the current command.
Before the result string of a measurement is read out, it is a good idea to check the internal event registers of the 3100 first. Most frequently, this check is a look at the test set's service register. Bit 2 of this register will be set as soon as there is an error message in the error queue.

NOTE

To avoid confusion between the registers of the STATus subsystem and the much more general SCPI statusbyte, the latter is called the service register throughout this manual.

The programming of the event handling system is described in detail in subsection "Understanding the STATus subsystem" on page 23.
The use of the service register is outlined in subsection "At the top: the service register" on page 25.

The scheme to use the registers for communication between a RAPID! program or the GPIB on one side and the 3100 on the other side always follows the following principles:

Step	RAPID!	GPIB
Preparing service register	`:PROGram:SRE <i1>`	`*SRE <i1>`
Set up a loop that waits for an event	`DO..."read service register"...LOOP`	`"SRQ"`
Read service register	`:PROGram:STB?`	serial poll or `*STB?`

Section "Programming examples" on page 16 shows some practical examples for the basic use of the register communication between SCPI and RAPID!.

Programming examples

In this subsection, you will find some application-oriented examples for RAPID! programs using SCPI on the 3100. These examples concentrate on the following topics:

- "Standard TX measurements" on page 17
- "Message exchange" on page 18
- "Example of a GPIB protocol" on page 19

Standard TX measurements

The example below illustrates how to perform a standard TX test using RAPID! and SCPI.

```
'FILE: RFTXDEMO.RBS
'DESCRIPTION: RAPID! program that measures
'the RMS phase error, the frequency,
'peak power and also checks whether the shape
'of the burst is within the PTT.

'Definition of variables
DIM result(5)

'Opening SCPI as communication port
LET scpi = freefile
OPEN "scpi" as #scpi

'Configuring the measurements as above as a group and
'starting a group measurement
PRINT #scpi, ":CONFigure:MEASure:GROup:RFTX
PRMS,FREQuency,POWer,TEMPlate"
PRINT #scpi, ":MEASure:RFTX:GROup?"

'Read out the result string
INPUT #scpi, result$

'Read out the service register to check if there was
'some error
PRINT #scpi, ":PROG:STB?"
INPUT #scpi, A$
A$ = BIN$(VAL(A$),8)

'Select reaction on event occurred
IF (MID$(A$,2,1) = "1" THEN
'Some error occurred
PRINT #scpi, ":SYST:ERR?"
INPUT #scpi, Err$
GOTO FAIL_EXIT
ELSE

'Measurements have been completed without errors
GOTO PRINT_RESULTS
END IF

'Result procedure of the program
PRINT_RESULTS:
'Converting the result string back into
'four result values
index = 0
result$ = result$ + ","

DO
```

```
' Find the position within result$, where the first
'comma appears
P = INSTR(result$, ",")
' Read out the part of result$ in front of the first
' comma and convert it into a numeric value
result(index) = VAL(LEFT$(result$, P - 1))
' Cut off the value just read from result$
result$ = MID$(result$, P +1))
index = index + 1
LOOP UNTIL result$=""

IF result(3) = 0 THEN
A$ = "PASS"
ELSE
A$ = "FAIL"
END IF

'Print results
PRINT "RMS phase deviation :"; result(0)
PRINT "Frequency :"; result(1)
PRINT "Peak power :"; result(2)
PRINT "Burst shape:"; A$
GOTO OKAY_EXIT

'Entry point of any error handling routine
FAIL_EXIT:
PRINT "An error did occur during measurements :", Err$

'Ending the program
OKAY_EXIT:
END
```

Message exchange

This example demonstrates a message exchange between a RAPID! program and an external control computer (connected on the GPIB).
The message exchange is triggered by events via the `SYSTem:MESSage` and the `PROGram:MESSage` queue.
Although not complete, the example below illustrates the principle.

NOTE

Only the core program code has been set in `courier` to make it easy to identify it. All comments are in 'normal' letters.

```
'FILE: MSGDEMO.RBS
'DESCRIPTION: RAPID! program to demonstrate message exchange between a
' RAPID! program and the controlling computer.

...

'Setting the mask for the service register: rising SRQ on an entry of a message
' into the message queue.
'Bits to set: SRQ (bit 6 = 64) and system.message (bit 0 = 1)
PRINT #0,"*SRE 65"
```

```
'Setting the mask for the program's service register: rising an event on an entry
' of a message. Bits to set: event (bit 6 = 64) and program.message (bit 0 = 1)
PRINT #0,":PROG:SRE 65"

DO

' Read out programm message queue
PRINT #0,":PROG:MESS?"
INPUT #0,message$

'Check message string for key letters
result$ = MID$(message$,1,2)
SELECT CASE result$
CASE "TX"
result$ = DoTXAlign(Message$)
CASE "RX"
result$ = DoRXAlign(Message$)
CASE "SP"
result$ = DoSPAlign(Message$)
CASE "AN"
result$ = DoAFANAlign(Message$)
CASE ELSE
result$=""
END SELECT

IF result$="" THEN PRINT #0,":SYST:MESS ";result$

LOOP
```

Example of a GPIB protocol

to/ from 3100	Data	Comment
to	:PROG:NAME MSG-DEMO.RBS	Load program
to	:PROG:STAT RUN	Start program on 3100
to	:PROG:MESS "TX, 1, 62, 124, -105"	Start (user-programmed) TX calibration process on 3100
from	SRQ	3100 raises service request after calibration
to	serial poll	Controller identifies SRQ device
from	SYST:MESS?	Controller queries system message
to	"TX Calibration OK"	3100 confirms successful TX calibration
from	:PROG:MESS "RX, 1, 62, 124, 15"	Start (user-programmed) RX calibration process on 3100

to/ from 3100	Data	Comment
	serial poll	Controller identifies SRQ device
to	`SYST:MESS?`	Controller queries system message
from	`"RX Calibration OK"`	3100 confirms successful RX calibration
to	`:MEASure:GSM:RFTX:ALL`	Start relevant RFTX measurements
	serial poll	Controller identifies SRQ device
to	`:CALC:GSM:RFTX:ALL:LIM:FAIL?`	Controller queries the result of the limit check of the measurement result values
from	`0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0`	3100 delivers results

Command subsystem overview

This subsection provides information about the SCPI command subsystems.

The Measurement subsystems – MEASure, FETCh, CALCulate

Measurement device configuration subsystems – RFGenerator, RFSPectrum, RFANalyser, AFGenerator, AFANalyser, MS Power Supply

The Node B and UE parameter subsystems – CONFigure, CALL

The communication-related subsystems – PROGram, STATus, SYSTem, FORMat

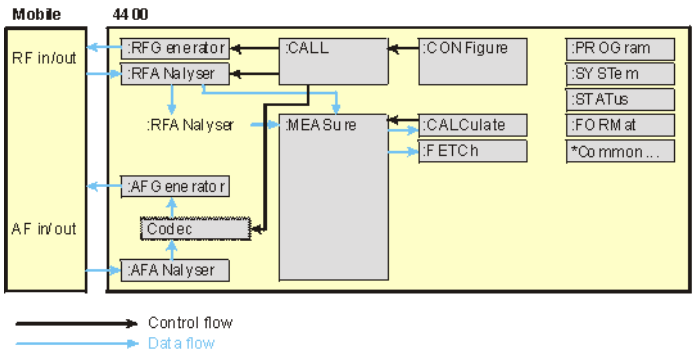
SCPI command errors

Using the SCPI commands

The various SCPI commands and their arguments/parameters are described below. Please note that any SCPI command specified with an invalid of without an argument required will be totally ignored by the system. This means that an incomplete SCPI command will not affect the current system status at all.

Schematic view of the subsystems of the 3100

This is a very simplified schematic view of the control and data flow inside the 3100.



Common commands

The common commands are understood by all SCPI and IEEE 488.2 instruments. Their purpose is to perform general tasks and to read or work with the registers common to all SCPI instruments. The following common commands have been implemented.

Command	Short description	Command group
:*CLS	Resets the entire status reporting system.	General common commands
:*ESE	Sets and queries the event status register mask.	Event status register
:*ESR?	Reads out the current contents of the event status register.	Event status register
:*IDN?	Returns device identity.	General common commands
:*OPC	Waits until previous command has been completed.	General common commands
:*RST	Resets the entire test set. All parameters, limits etc. will be set to internally predefined default values.	General common commands
:*SRE	Sets and queries service register mask.	Service register
:*STB?	Returns the current contents of the service register and clears the service register.	Service register
:*WAI	Waits until previous command has been completed.	General common commands

The communication-related subsystems

These subsystems cover system relevant tasks, which are not primarily involved in the measurement process.

SYSTem – System parameters, such as the number of unread error messages, the test set's GBIP address etc. may be read out or set using the commands of this subsystem.

PROGram – This subsystem deals with all activities related to RAPID! programs.

STATus – The STATus subsystem controls and provides information on the state of the 3100. There are two types of states: operational states describe what is currently going on within the test set while questionable states deliver questionable states of the 3100.

FORMat – The commands of this subsystem enable settings of the data output format in remote mode.

The SYSTem subsystem

System parameters, such as the number of unread error messages, the test set's GPIB address etc. may be read out or set using the commands of this subsystem.

The STATus subsystem

The STATus subsystem delivers detailed information about the internal status of the 3100, its error conditions and special events. These three areas are dealt with in three different sections of the status subsystem. These three sections are addressed, using different commands.

Status Area	Related commands	Main functional aspects
Operation Status	STATus:OPERation...	These commands deal with the operation status of the 3100. They describe what is currently going on inside the test set, mainly in respect of signaling and measuring.
System Errors	STATus:QUEStionable...	This area of the internal status report system mainly deals with errors and warnings regarding the hardware stages of the 3100 (like 'RF input overload' or 'frequency out of range'.
Execution Errors	*ESE, *ESR?	Mainly program or SCPI command execution errors are dealt with in this area of the internal status report system.

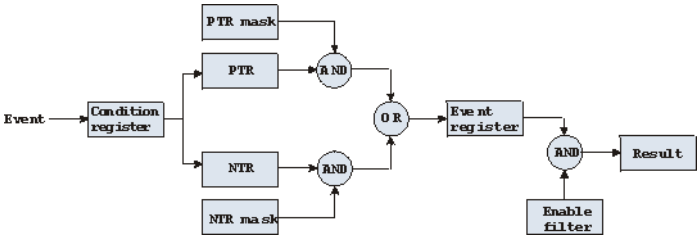
The status subsystem provides in-depth information about the internal status of the test set. Furthermore, powerful event processing tools allow any form of flexible control over the 3100.
However, the use of the status subsystem is a bit tricky because of the many parameters involved. Therefore, we kindly suggest to carefully read subsection "Understanding the STATus subsystem" on page 23 before using the status subsystem.
A table of all registers implemented can be found in subsection "Table of registers" on page 26.

Understanding the STATus subsystem

The STATus subsystem is intended to deal with special events occurring inside the test set. It provides tools to enable any current condition to raise a system event. These system events may then be used to trigger service requests (SRQ) on GPIB, or to trigger RAPID! programs.

There are several **groups of registers**, structured in a hierarchical order. Lower-level registers work on specific conditions and single parameters while higher-level registers integrate the result of several lower-level registers and thus provide a more general view.
The highest level of these registers is the **service register**, sometimes also referred to as the **statusbyte register** or **status byte**.

The figure below gives an example of a group of registers.



A group of registers starts with a **Condition register**. An internal **Event** will set the corresponding bit of the Condition register (to 'set a bit' means a transition from binary **0** to a binary **1**). Any Condition register will be updated continuously. This means that a bit will be reset as soon as the condition which rose that bit is no longer valid.

Example:

As soon as the 3100 starts to page a mobile under test, a certain bit of a certain condition register will be set. When the 3100 stops paging the mobile (because e.g. it responded to the paging requests), this bit will be reset. Now, there is no chance to find any evidence for a paging process in this Condition register.

Therefore, there is an **Event register** included in every group of registers. In the Event register, a bit will remain set even when the condition for it to be set is no longer valid. Any Event register, however, will be cleared after a query.

Example:

Continuing our example from above, in the related Event register, the corresponding bit would still be set. A query of this register would provide evidence that there has been some paging in progress. However, the Event register is not able to provide any information if the condition is still valid.

Summary 1: condition-type versus event-type registers

- The condition-type registers reflects the current status of the test set and is updated continuously. When you would like to know if a special condition is currently valid, then read out the related condition-type register with a query.
- The condition-type register and the event-type register have an identical structure. This means that they are of the same length and the single indicator bits are at the same positions.
- The event-type registers are the 'memory' of the status system. Once a bit has been set, it will remain set until the event-type register is read out with a query. When you want to trigger e.g. a RAPID! program with a certain event, always read out the related event-type register.
- Event-type registers are read-only and self-destructive. They will be cleared after any query.

Transition of a bit from the condition-type to the event-type register

How will a bit find its way to the **Event register**?
This depends on the transition filter and its settings. The transition filter works as follows:

First, there are two branches: the **P**ositive **TR**ansition and the **N**egative **TR**ansition filter. Both only react on the corresponding transitions of bits and both contain as many bit positions as the condition register.

PTR will carry a binary 1 at a bit position only when the corresponding bit of the condition register is **set**, while **NTR** will carry a binary 1 at a bit position only when the corresponding bit of the condition register is **reset**..

Both the output of PTR and NTR will be combined with the corresponding mask, using a logical **AND** operation.

These masks are user-definable (using the **`STAT:xxxx:xTR`** commands) and again contain as many bit positions as the condition register.

Example:
The only chance for bit 4 (that has just been **set** in the condition register) to reach the **Event register** is that the **PTR mask** carries a binary **`1`** at bit position 4. The logical AND between the PTR filter and the PTR mask will then deliver a binary **`1`**. This binary **`1`** will pass the logical OR and thus set bit 4 of the **Event register**.

Summary 2: from the condition-type to the event-type registers

- There are two detectors for every bit of a condition register: the **positive** transition and the **negative** transition branch. Positive transition means a change of a bit from a binary **0** to a binary **1** while a negative transition is a bit change from a binary **1** to a **0**.
- To every branch, there is a filter mask (PTR and NTR mask). This filter mask is user-definable.
- The filter mask and the result of the transition filter are combined using a logical AND operation.
- The results of the AND operation in both branches will be combined, using a logical OR operation.
- The results of the OR operation are the contents of the related event-type register.

Moving up: an event reaches the condition register one hierarchical level up

The **Event register** contains summary bits, corresponding to the bits of the condition register. A summary bit will be set, when the initial event passes the transition filter.

The **Enable filter** is a mask to filter events that are allowed to move one level up. The Enable filter is user-definable (using the **`STAT:xxxx:ENABle`** commands) and again contains as many bit positions as the condition-type register.

Again, the Enable filter mask is ANDed with the Event register and a nonzero result will finally set the **Result** bit. This Result bit may be a summary bit either in a higher-level register, or in the service register.

Summary 3: raising the Result bit

- The event-type register will be ANDed with the related Enable filter mask.
- The Enable filter mask is user-definable.
- The result of the logical AND operation between event-type register and related Enable filter will be the Result bit.
- The result bit will set the corresponding bit of the condition register one level up.

At the top: the service register

The **service register** contains eight summary bits: three for the status groups available on the 3100, two for internal queue handling, two for remote control and one bit for it's own status: the **summary status bit**.

When an event passes through and sets one of the seven corresponding summary bits of the service register, the contents of the service register will be ANDed with the service register mask. This mask can be set using the **`*SRE`** common command.

When the logical AND operation of the current contents of the service register and the service register mask leads to a binary **1**, then the summary status bit will be set as well.

If both bit 6 of the service register mask and bit 6 (the summary status bit) of the service register are set, then a service request (SRQ) is executed.

Table of registers

The STATus subsystem uses and/or provides access the following registers:

Service register

This register represents the highest level within the report structure of the 3100. The service register contains eight bits. A detailed description of the service register can be found in the appendix (SCPI Command Reference).

Event status register group

This group of registers collects all general events of the 3100 (mostly command errors).
Depending of the setting of the event status register mask, bits set in the event status register may be transferred to bit 5 of the service register.
For further details regarding the event status register, please refer to the appendix (SCPI Command Reference).

General operation register group

This group of registers is 16 bits wide and reflects the general operation status of the 3100.
Some of the bits of this register (like bits 8 and 9) are summary bits. Those summary bits are result bits of subordinate groups of registers as described below.

NOTE

The commands related to the general operation register group and its subordinate groups of registers all start with **`:STATus:OPERation:`**.

Bit	Decimal	Meaning
0	1	Set during calibration.
1	2	Set while settling.
2	4	Set while ranging.
3	8	Set while sweeping.
4	16	Set as long as any measurement is being carried through.
5	32	Set while the 3100 waits for a trigger.
6	64	Set while the 3100 waits for an arm.
7	128	Set while a correction process is in progress.
8	256	This is the SIGNaling summary bit. This is the Result bit of the signaling operation register group (see below for details).
9	512	This is the MEASure summary bit. This is the Result bit of the measuring operation register group (see below for details).

10	1024	Not in use.
11	2048	Not in use.
12	4096	Not in use.
13	8192	Instrument.
14	16384	Not in use.
15	32768	Not in use.

Signaling operation register group

This group of registers is 16 bits wide. Its main task is to deal with events related to the signaling status. The signaling operations depend on the system option and the contents of the signaling operation register group are detailed in the appendix.
The Result bit of this group of registers is forwarded to bit 8 of the general operation register group.

Measuring operation register group

This group of registers is 16 bits wide. Its main task is to deal with events related to the measurement status.
The Result bit of this group of registers is forwarded to bit 9 of the general operation register group.

Bit	Decimal	Meaning
0	1	Set while RFTX measurements are in progress.
1	2	Set while RFRX (BER) measurements are in progress.
2	4	Set while RFSPectrum measurements are in progress.
3	8	Set while audio measurements are in progress.
4	16	Not in use.
5	32	Not in use.
6	64	Not in use.
7	128	Not in use.
8	256	Not in use.
9	512	Not in use.
10	1024	Not in use.
11	2048	Not in use.
12	4096	Not in use.
13	8192	Not in use.
14	16384	Not in use.
15	32768	Not in use.

General questionable status register group

This group of registers is 16 bits wide and reflects the general questionable status of the 3100. The events taken care of this group of registers are mainly errors and warnings.

Some of the bits of this register (like bits 9,10 and 11) are summary bits. Those summary bits are result bits of subordinate groups of registers and described below.

NOTE

The commands related to the general questionable status register group and its subordinate groups of registers all start with **`:STATus:QUEStionable:`**.

Bit	Decimal	Meaning
0	1	Voltage out of range. Not used on the 3100.
1	2	Current out of range. Not used on the 3100.
2	4	Time out of range. Not used on the 3100.
3	8	Power out of range. Not used on the 3100.
4	16	Temperature out of range. Not used on the 3100.
5	32	Frequency out of range. Not used on the 3100.
6	64	Phase out of range. Not used on the 3100.
7	128	Modulation out of range. Not used on the 3100.
8	256	Calibration out of range.
9	512	This is the RF summary bit. This is the Result bit of the RF questionable status register group (see below for details).
10	1024	This is the SYNChronization summary bit. This is the Result bit of the SNYChronization questionable status register group (see below for details).
11	2048	This is the AUDio summary bit. This is the Result bit of the AUDio questionable status register group (see below for details).
12	4096	Not in use.
13	8192	General warning, concerning the test set.
14	16384	Command not understood warning.
15	32768	Not in use.

RF questionable status register group

This group of registers is 16 bits wide. Its main task is to deal with warnings and errors regarding the RF stages of the 3100.
The Result bit of this group of registers is forwarded to bit 9 of the general questionable status register group.

Bit	Decimal	Meaning
0	1	Input overload. Reduce RF power immediately to avoid possible damage of the 3100's highly sensitive input stages!
1	2	Output level out of range.
2	4	Transmission frequency out of range.
3	8	Reception frequency out of range.
4...15		Not in use.

SYNChronization questionable status register group

This group of registers is 16 bits wide. Its main task is to deal with warnings and errors regarding the external synchronization of the 3100.
The Result bit of this group of registers is forwarded to bit 10 of the general questionable status register group.

Bit	Decimal	Meaning
0	1	Set when an external RF synchronization signal is recognized on the EXT SNYC prog. socket.
1	2	Set when an external frame synchronization signal is recognized on the SNYC IN/OUT socket.
2...15		Not in use.

AUDio questionable status register group

This group of registers is 16 bits wide. Its main task is to deal with warnings and errors regarding the audio stages of the 3100.
The Result bit of this group of registers is forwarded to bit 11 of the general questionable status register group.

Bit	Decimal	Meaning
0	1	Input overload. Reduce signal level immediately to avoid possible damage of the 3100's highly sensitive input stages!
1	2	Output level out of range.
2...15		Not in use.

The PROGram subsystem – overview

This subsystem contains commands related to loading and executing RAPID! program files.

The FORMat subsystem – overview

The FORMat subsystem sets and queries settings concerning the data output in remote mode.

The Node B and UE parameter subsystems

These subsystems allow access to the node B parameters (i.e. the 3100 simulating a node B) and to the information received from the mobile under test like the measurement report. The commands of these subsystems are described here.

CONFigure – This subsystem incorporates all changeable node B parameters of all implemented communication systems.

NOTE

The settings made here directly affect all communication system-specific subsystems.

CALL – This subsystem handles call procedures and allows to read out the measurement report generated by the mobile under test.
The information available is dependent on the current state of a call, i.e. some commands require an established radio communication link between the 3100 and the mobile under test.

The CONFigure subsystem

This subsystem incorporates all changeable NB parameters of all systems implemented. The key commands are the following:

CONFigure:CSYStem	Selects the communications system to work with.
CONFigure:<SystemOption>:...	These commands select parameters within a communications system.
CONFigure:<SystemOption>:NB:...	These are the commands to set specific system parameters like the node B's RF output power level or its identity.
CONFigure:<SystemOption>:UE:...	The mobile-specific information is handed over to the 3100 using these commands. One example is the power level.
CONFigure:<SystemOption>:BER:...	These commands set the BER parameters.
CONFigure:<SystemOption>:GROup:...	With the help of these commands, groups of measurements may be defined.
CONFigure:COUPloss:...	These commands provide access to the coupling loss compensation feature of the 3100.

NOTE

Always select the communication system via the `:Configure:CSYStem` command first before you change any parameters using different SCPI commands.

The CALL subsystem

This subsystem contains commands

- for call setup and handling procedures and
- to read out the measurement report, generated by the mobile.

The Measurement subsystems

The 3100 provides the following measurement subsystems:

The MEASure subsystem – This subsystem provides the commands for all kinds of measurements: Single-shot as well as series of measurements, measurements of single parameters as well as of groups of parameters.

The FETCh subsystem – To read out the lastest measurement result of a specific parameter or a group of parameters.

NOTE

FETCh requires that a measurement is started first, using the MEASure commands.

NOTE

FETCh will neither start nor terminate continuous measurements.

The CALCulate Subsystem – All kinds of statistic evaluations and checks of measurement results against predefined limits.

The MEASure subsystem

The MEASure subsystem is probably the most important SCPI command subsystem of the 3100. There you will find all commands to acquire measurement results of the mobile under test.
Measurements can be taken as one-shot measurements or as series of measurements:

Type of measure-ment	Related command element	Short description
One-shot	[:CONTinuous]	Actually `:CONT` starts a measurement, that will be performed continuously. Single measurement results can be read out using the related `:FETCh` commands. Thus several measurements can be started (where those measurements are started first that take the longest time, like e.g. BER).
Series	:ARRay	This command element offers the possibility to carry through a specific measurement a certain number of times. All the single measurement result values can be read out with just one command. Using this feature, measurement results returned can then be used e.g. for statistic data evaluation on an external computer.

Example:

First a BER measurement command is issued to start a BER measurement (because it takes some time).
Then several RF TX measurements are performed and the measurement results are read out and used for statistic data evaluation.
After that the result of the BER measurement is read out.

Important notes:

- A newly issued **RF** TX MEASure command will terminate any **RF** TX or **RF** spectrum measurements currently in progress.
- A newly issued **RF** SPectrum MEASure command will terminate any **RF** TX or **RF** spectrum measurements currently in progress.
- A newly issued RF RX MEASure command will terminate any RF RX measurements currently in progress.
- A newly issued AF MEASure command will terminate any RF TX, RF RX, RF spectrum or AF measurements currently in progress.
- Measurements are always started, using the current system parameters and the current state of the 3100.
- In case the 3100 is in a state that does not allow a specific measurement command to be performed or completed, an error message will be added to the 3100's internal error queue. The related flag in The STATus subsystem will be set, too.
  In case this is the first error message to appear within the error queue, the error indicator bit of the service register (bit 2) will be set as well.

:MEASure[:CONTinuous] MEASure commands that are stated with or without the optional `:CONTinuous` command element will make the 3100 perform the related measurement for an unlimited number of times.
The measurement will only be terminated if a new MEASure command of the same or related type is issued (see **Notes** above).

Syntax	`:MEASure[:CONTinuous]<{:measProp}>[?]`
Parameters	The one-shot measurements do not require any parameters.
Description	Starts the (continuous) measurement of the specified measurable property. The `CONTinuous` command element is optional. Unless a measurement result is read out (using the `FETCh` subsystem), no measurement result values will be given back. The measurement result(s) of any measurement will be stored internally. Any previously stored result will be overwritten as soon as a new measurement result has been achieved. The latest result measured may be read out using the FETCh subsystem. Any FETCh command will wait for a measurement result value(s) to be available. In case there is more than one measurement result value, the single measurement result values are separated by commas (like e.g. "50.5,3.46"). Should the FETCh command fail to obtain a measurement result value (because e.g. the current state of the 3100 does not allow the measurement to be performed or completed), a timeout will occur and an error message will be added to the 3100's internal error queue. The related flag in the STATus subsystem will be set, too. In case this is the first error message to appear within the error queue, the error indicator bit of the service register (bit 2) will be set as well. The main application of the combination of the MEAS and the FETC subsystems: Starting a measurement that takes some time to deliver a measurement result back (e.g. the BER measurement. After the measurement has been started with the MEAS command, the test set 'is free' to perform other tasks. If in this case the query form of the MEAS command is used, the test set is blocked until a measurement result is available. Sometimes, measurement results will need to 'sweep in'. In this case, the first measurement result might be totally misguiding. Using the `MEAS` command will allow the measurement result value to stabilize on a meaningful result before the latest result value is read out using the `FETC` subsystem.
Query	The query form of any MEASure command will start the (continuous) measurement of the specified measurable property. The `CONTinuous` command element is optional. After the first measurement has been completed, the measurement result value(s) will be delivered back in a string as outlined above. Should the query fail to obtain a measurement result value, a timeout will occur and an error message will be added to the 3100's internal error queue as explained above. The main application of the query form of a `MEAS` command is speed. When combined with 'fast' measurements (like e.g. the fast power level measurement), the query form of a `MEAS` command delivers measurement results as fast as possible. **Note:** The measurement started with the query form of the `MEAS` command will be continued in the background. Any further measurement result values may be read out, using the appropriate `FETCh` command.

Examples

`:MEASure:GSM:CONTinuous:RFTX:PPEAk` and
`:MEASure:GSM:RFTX:PPEAk`
are identical. Either command will start the measurement of the peak phase error. The latest result of this measurement (like "`5.84`") will be stored internally. It will be overwritten as soon as a new measurement result has been achieved.
To read out the current measurement result, use the `FETCh` subsystem:
`:FETCh:GSM:RFTX:PPEAk?`

:MEASure:RFTX:ALL
This command will start the continuous measurement of all relevant RFTX parameters. After this command has been issued, you may continue with e.g. an RF RX MEASure command.
All 19 single measurement results will be stored internally. As soon as a new measurement result has been achieved, the previous value will be overwritten.
To read out the measurement results achieved by the `:MEAS:RFTX:ALL` command, use the `:FETCh:GSM:RFTX:ALL?` command.

:MEASure:GSM:RFTX:ALL?
As in the example above, the continuous measurement of all relevant RFTX parameters will be started. Unlike the example above, this command will wait until all 19 measurement results have been achieved and will return all of them in a string, separated by commas. The measurements will continue and later results may be read out, using the `FETCh:GSM:RFTX:ALL?` command.

How to convert a result string back into numeric variables

The example program below illustrates how the returned string can be converted back into numeric variables in a RAPID! program.

```
PRINT #scpi, ":MEAS:GSM:RFTX:ALL?"
INPUT #scpi, result$
index = 0
result$ = result$ + ","

DO

' Find the position within result$, where the first
comma appears
P = INSTR(result$, ",")
' Read out the part of result$ in front of the first
comma and convert it into a numeric value
result(index) = VAL(LEFT$(result$, P - 1))
' Cut off the value just read from result$
result$ = MID$(result$, P +1))
index = index + 1

LOOP UNTIL result$=""
```

MEASure:ARRay

The `:ARRay` command element makes the 3100 perform any measurement property a user-definable number of times. All measurement result values obtained during the process will be stored in an internal array and can be read out using the related commands of the FETCh subsystem or will be returned in case the measurement process has been started using the query format of the command.
After the specified number of measurements have been performed, the measurement will be stopped and no further measurement result values will be stored

internally. Therefore, any attempt to read out data again (unless any measurement has been started before) will result in a timeout and thus in an error message.
This is one of the main differences between the `[:CONTinuous]` and the `:ARRay` command element.

Note: Any measurement will be terminated if a new MEASure command of the same or related type is issued (see **Notes** in section "The MEASure subsystem" on page 31 for details).

Syntax	`MEASure:ARRay<{:measProp}>[?] <numMeas>`
Parameters	`numMeas` is the number of measurements to be performed.
Description	Takes a `numMeas` number of measurements of the specified type `measProp`. The results of the single measurements will be stored in an internal array. The measurement results array can be read out using the related command of the FETCh subsystem (see examples below for reference). After an array has been read out using the related `FETCh` command, the internal array will be cleared. Any subsequent `FETCh` command trying to read out the same measurement result array will not be able to read any measurement results and thus result in a timeout. In this case, an error message will be added to the 3100's internal error queue (for further details, refer to section ":MEASure[:CONTinuous]" on page 33).
Query	The query form of any `MEAS:ARR` command will start the related measurement the specified number of times. After the number of measurements specified have been performed, the measurement will be stopped. The query will then return a string containing all the measurement results. The single result values will be separated by commas. Any subsequent `FETCh` command trying to read out the same measurement result array will not be able to read any measurement results and thus result in a timeout. In this case, an error message will be added to the 3100's internal error queue (for further details, refer to section ":MEASure[:CONTinuous]" on page 33).

Examples	`:MEASure:GSM:ARRay:RFTX:PPEAk 10`
	This command will make the 3100 perform 10 independent measurements of the maximum phase error in GSM.
	After those 10 results have been achieved, the measurement will be stopped.
	The ten result values will be stored in an internal array.
	To read out the measurement result array, use the `FETCh` subsystem:
	`:FETCh:GSM:RFTX:PPEAK?`
	will return the 10 values in one string (like
	`"5.42,5.44,5.80,...5.72,5.64"`)
	Any subsequent `:FETCh:GSM:RFTX:PPEAK?` command will result in a timeout.
	`MEASure:GSM:ARRay:RFTX:ALL? 2`
	This command takes all relevant RFTX measurements twice. The measurements will be stopped as soon as the 2 x 19 result values are available.
	The 38 result values will be returned as a string; the single values will be separated by commas.
	Any subsequent `:FETCh:GSM:RFTX:PPEAK?` command will result in a timeout.

How to convert a result string back into numeric variables

The example program below illustrates how the returned string can be converted back into numeric variables in a RAPID! program.

```
PRINT #scpi, ":MEAS:GSM:ARR:RFTX:ALL? 2"
INPUT #scpi, result$
index = 0
result$ = result$ + ","

DO

' Find the position within result$, where the first
comma appears
P = INSTR(result$, ",")
' Read out the part of result$ in front of the first
comma and convert it into a numeric value
result(index) = VAL(LEFT$(result$, P - 1))
' Cut off the value just read from result$
result$ = MID$(result$, P +1))
index = index + 1

LOOP UNTIL result$=""
```

:MEAS[:CONT]:BLOCkdata

The `BLOCkdata` command element of this subsystem is used to read out all the single measurement results necessary to generate the following result graphics:

- shape of the TX burst (burst received by the 3100 from the mobile)
- phase error
- modulation spectrum of burst received by the 3100 from the mobile
- AF spectrum (the audio spectrum)

:MEASure:BLOCkdata:BURStshape

This command starts a continuous measurement of the mobile's burst.
The corresponding result string (to be read out either with the query form of the command or with `:FETCh:BLOCkdata:BURStshape` will contain 711 floating point real numbers, representing the measurement result values. The numbers will only have one digit behind the dot. The single measurement result values are separated by commas.
The meaning of the single values within the result string is as follows:

- The first two values are necessary for the scaling of the data. All remaining 709 values represent the measurement results.
- The first value (index = 0) gives the offset of the middle of the burst (bit 73) in respect to the measurement result array.
  Example: An offset of `353.0` means that the middle of the burst will be the 355^{th} value within the array (or the value contained at index 354).
- The second value (index = 1) gives the peak power level of the burst in dBm.
- All following values (indices 2 to 710) give the RF power levels of the related positions.

The resolution on the time axis is approx. 0.875 microseconds (i.e. the distance in time between two consecutive measurement points).

:MEASure:BLOCkdata:PHASeerror

First, a continuous measurement of the mobile's phase error will be started.
The corresponding result string (to be read out either with the query form of the command or with `:FETCh:BLOCkdata:PHASeerror` will contain 711 floating point numbers, representing the measurement result values. The single measurement result values are separated by commas.
The meaning of the single values within the result string is as follows:

- The first two values are necessary for the scaling of the data. All remaining 709 values represent the measurement results.
- The first value (index = 0) gives the offset of the middle of the burst (bit 73) in respect to the measurement result array.
  Example An offset of `353.0` means that the middle of the burst will be the 355^{th} value within the array (or the value contained at index 354).
- The second value (index = 1) is always set to `0.0`.
- All following values (indices 2 to 710) give the phase error of the related positions.

The time resolution is approx. 0.25 bit period (i.e. four measurement results per bit period).

:MEAS:BLOC:MSP[:CURRent]

This command starts a continuous measurement of the modulation spectrum of the mobile's burst.
The corresponding result string (to be read out either with the query form of the command or with `FETCh:BLOCkdata:MSPectrum` will contain floating point real numbers, representing the measurement result values. The single measurement result values are separated by commas.
The number of the measurement result values returned is dependent on the span and resolution, set with the commands `:RFSPectrum:MSPectrum:SPAN` and `:RFSPectrum:MSPectrum:RESolution`.

:MEASure:BLOCkdata:MSPectrum:AVG

This command starts the measurement of the modulation spectrum of the mobile's burst for a user-definable number of times and averages the measurement results.
The corresponding result string (to be read out either with the query form of the command or with `FETCh:BLOCkdata:MSPectrum` will contain floating point real numbers, representing the measurement result values. The single measurement result values are separated by commas.
The number of the measurement result values returned is dependent on the span and resolution, set with the commands `:RFSPectrum:MSPectrum:SPAN` and `:RFSPectrum:MSPectrum:RESolution`.

:MEASure:BLOCkdata:AFSPectrum

This command starts a continuous measurement of the audio spectrum.
The corresponding result string (to be read out either with the query form of the command or with `FETCh:BLOCkdata:AFSPectrum` will contain floating point real numbers, representing the measurement result values. The single measurement result values are separated by commas. The number of the measurement result values returned is dependent on the span and resolution.

NOTE

Audio measurements can only be performed on the 3100 when the Audio Option is installed.

:MEASure:...:GROup

As mentioned before (see **Notes** in section "The MEASure subsystem" on page 31), a new measurement will always terminate a preceding one of the same or related type.
Therefore, the `GROup` command element has been implemented in the SCPI command set of the 3100. This command element allows to specify a user-definable list of measurements than can then be started with one command. The measurement results can be read out using the query form of this command – or with the related command of the FETCh subsystem..

NOTE

In this respect, `:MEAS:RFTX:ALL` can be regarded as a predefined 'group', containing all important RFTX measurements.

NOTE

The `AFANalyser` subsystem also allows to define 'groups'. However, all commands of the `AFANalyser` subsystem will only obtain measurement results if the Audio Option has been installed.

Syntax

`:CONFigure:MEASure:GROup[:RFTX] <{RFTXprop}>`
`MEASure[:CONTinuous]:RFTX:GROup` **or**
`MEASure:ARRay:RFTX:GROup`
or
`:CONFigure:MEASure:GROup:AFANalyser <{AFANprop}>`
`MEASure[:CONTinuous]:AFANalyser:GROup` **or**
`MEASure:ARRay:AFANalyser:GROup`

Parameters

`<{RFTXprop}>` **is one or more of the single RFTX measurements**
`PPEAk,PRMS,FREQuency,LENGth,UTIMe,POWer,TEMPlate,CORNer,FLATness`
`<{AFANprop}>` **is one or more of the single AF Analyser measurements**
`SINad,DISTortion,FREQuency,ACV:PEAKp,ACV:RMS,DCV:RMS`

Description

Starting a 'group' measurement will take all measurements specified just with one single command.
After the group command has been completed, all 'group' measurement results are available at the same time and can be read out using either the query form of the command or the related commands of the FETCh subsystem.

Example

```
:CONF:GSM:MEAS:GRO:RFTX PPEAK,FREQency,POWer,LENGth
:MEAS:RFTX:GRO
```

This sequence of commands first defines a group of RFTX measurements and then issues a group command. As soon as all the measurements specified in the group command have been completed, the measurement result values can be read out using the `:FETCh:GSM:RFTX:GRO?` **command.**

The FETCh subsystem

The FETCh subsystem enables you to read out the currently valid measurement result value(s) of a measurement.

Important notes

- Before a measurement value may be read out with commands of the FETCh subsystem, a MEASure command has to be issued first.
- After a continuous measurement has been started, the latest measurement result value can be obtained using the related **`:FETCh`** command.
- In case an array measurement has been started, the related **`:FETCh`** command will return the entire measurement result array.
- If there are no measurement results to be read out by a **`FETCh`** command for any reason, a timeout will occur. The wait time until a timeout occurs is dependent on the type of measurement to be performed (see below).
- If the preceding **`MEASure`** command and the **`FETCh`** command do not match, a timeout will occur.
- When you use the query form of any **`MEASure`** command, all measurement results obtained will be handed back and the internal result register will be cleared afterwards. Consequently, a subsequent **`FETCh`** command will lead to a timeout (as above).
- The following timeouts have been implemented:
  - 5 s for all RFTX measurements
  - 30 s for all RFRX measurements
  - 10 s for all RFSPectrum measurements
  - 10 s for all AF measurements

There are two versions of a FETCh command:

- The **`:FETCh:LAST?`** command will read out the latest result of the last MEASurement command issued - whatever command that was.
  Using this command, please keep in mind that your control program then has to take care of the number and format of the measurement result values returned.
- The **`:FETCH:{measProp};`** commands will read out the latest result of the measurement specified with **`{measProp}`**.

FETCh:LAST

The **`:FETCh:LAST?`** command will read out the latest result of the last MEASurement command issued - whatever command that was.
Using this command, please keep in mind that your control program then has to take care of the number and format of the measurement result values returned.
To convert a result string back into single measurement result values, please refer to section "How to convert a result string back into numeric variables" on page 34.

Syntax	**FETCh:LAST**?
Description	Returns a string, containing the latest measurement result(s) for the **MEASure** command issued last. Format and number of the measurement result values contained in the string are depending on the preceding **MEASure** command.
Examples	**MEASure:GSM:RFTX:PPEAk** **FETCh:LAST?** will return the latest result (e.g. **5.84**). **MEASure:GSM:ARRay:RFTX:PPEAk 5** **FETCh:LAST?** will return the entire array (e.g. **5.84,5.81,5.94,5.74,5.79**). **MEASure:GSM:ARRay:RFTX:PPEAk? 5** **FETCh:LAST?** In this case, a timeout will occur as the **:MEAS** query will return the entire array and there will be no measurement result values for the **:FETCh** command to collect.

FETCh:BLOCkdata:...? The commands with the **:BLOCkdata** command element are used to read out all the single measurement results necessary to generate the following result graphics:

- shape of the TX burst (burst received by the 3100 from the mobile)
- phase error
- modulation spectrum of burst received by the 3100 from the mobile)
- AF spectrum (the audio spectrum)

FETCh<{:measProp}> The **:FETCH<{:measProp}>** commands will read out the latest result of the measurement specified with **measProp**.

Syntax	**FETCh**<{:measProp}>?
Description	Returns a string, containing the last measurement result(s) obtained for **measProp**.

Examples

`MEASure:GSM:RFTX:ALL`
The result of this command will be that the 3100 continuously measures all RFTX parameters.
To read out the 19 measurement result values, use the
`FETCh:GSM:RFTX:ALL?`
command. It will deliver back all measurement results contained in a string. This command may be used repeatedly (as long as the corresponding **`MEASure`** command was not terminated by another **`MEASure`** command.

`MEASure:GSM:ARRay:RFTX:PPEAk 10`
This command will start the measurement of the peak phase error. As soon as 10 measurement results have been obtained, the measurements will be stopped.
To read out the 10 measurement results, use the
`FETCh:GSM:RFTX:PPEAk?`
The string delivered back contains the 10 measurement results, starting from the first one (the oldest).

`MEASure:GSM:ARRay:RFTX:PPEAk? 10`
Unlike the example above, this command will start the measurement of the peak phase error. As soon as 10 measurement results have been obtained, those results will be given back in a string and the measurements will be stopped.
Any
`FETCh:GSM:RFTX:PPEAk?`
command will lead to a timeout as there are no new measurement results to be collected.

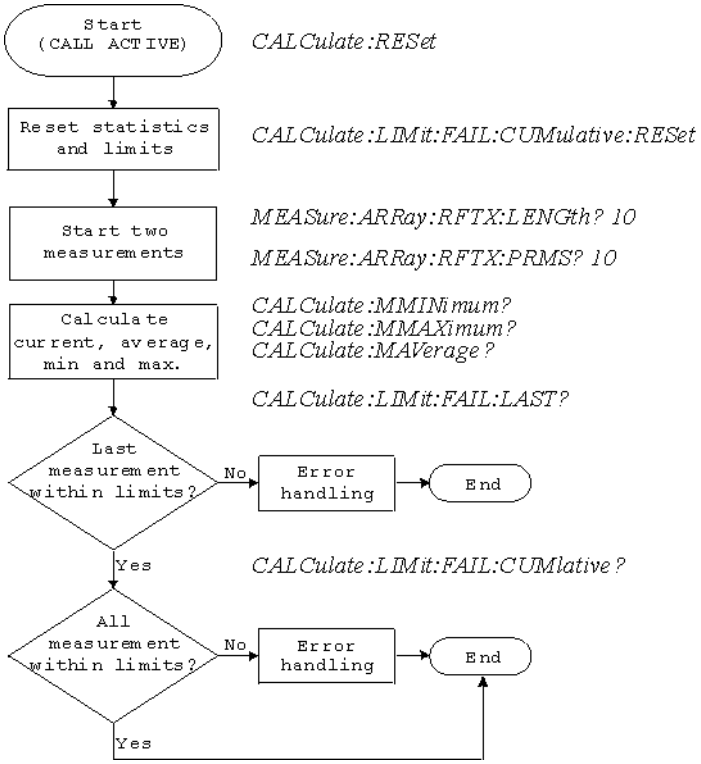
The CALCulate Subsystem

The CALCulate subsystem performs statistic evaluation of measurement results and also allows to check results against user-definable limits.

The basic scheme outlined below gives an idea of how to work with the CALCulate subsystem and to use the single queries as branching decisions within a program flow.

NOTE

Statistic evaluations on measurable properties like average, minimum or maximum will only be performed on the type of measurement started last (i.e. the **`PRMS`** measurement in the example below).



Reading the basic scheme

- First, the limits of the measurements to be performed are set (not shown in the basic scheme).
- After a call has been established, the CALCulate subsystem is reset using the **`:CALCulate:RESet`** command. Only measurements started after this command will be taken into account for the statistic evaluation. Any results from previously started (and still running) measurements will be ignored as well as all existing measurement results.
- The CALCulate subsystem of the 3100 allows to perform a cumulative check of the measurement results of a measurement. This means that all measurement results obtained (since the last reset of this part of the subsystem) will be checked against the corresponding limits.
  This cumulative check is reset as well in the example shown above with the **`:CALCulate:LIMit:FAIL:CUMulative:RESet`** command.
  For further details, check with section "CALCulate:LIMit" on page 43.
- Then, two measurements are started in this example (commands **`:MEASure:ARRay:RFTX:LENGth? 10`** and **`:MEASure:ARRay:RFTX:PRMS? 10`**).
- As soon as the measurements have been completed, the CALCulate subsystem is used to identify the minimum and maximum measurement result value and to calculate the average measurement result value. Please note that these commands will work **only** on the results of the measurement started last (the RMS-valued phase error in this example).
- The command **`:CALCulate:LIMit:FAIL:LAST?`** will return a boolean number, indicating whether the last measurement of the RMS-valued phase error was within the limits set (then a **`0`** will be returned) or whether it was off the limits (then a **`1`** will be returned.)

– The last command (**:CALCulate:LIMit:FAIL:CUMulative?**) is similar to the one above, with the main difference that this query will tell whether **all** measurements of the RMS-valued phase error taken since the last reset of the cumulative check are within the limits set (then a **0** will be returned) or whether at least one measurement result value did violate at least one of the user-definable limits (in this case, the query will deliver back a **1**).

CALCulate:LIMit

The **CALCulate** commands that incorporate the **LIMit** command element

– check whether one or more measurement result(s) did violate the user-definable limits
– reset the cumulative limit evaluation system
– switch the limit evaluation system for specific measurements on or off
– set the limits

FAIL[:LAST]? – did the latest measurement result value fail?

Syntax	**:CALCulate:LIMit:**FAIL[:LAST]?
Returns	**0** when the limits were not violated or **1** in case at least one limit was violated by the latest measurement result value.
Description	Checks whether the **latest** measurement result value of the measurement started last is within its limits.
Examples	**:MEAS:GSM:ARR:RFTX:LENG** **:MEAS:GSM:ARR:RFTX:PRMS** **:CALC:GSM:LIM:FAIL:LAST?** This command of the CALCulate subsystem will check whether the latest measurement result value of the **PRMS** measurement is within its limits. **:MEAS:GSM:ARR:RFTX:LENG 10** **:CALC:GSM:LIM:FAIL:LAST?** In this example, the **:CALC** command will check if **all 10** measurement result values of the burst length measurement array are within their limits.

{measProp}:LIMit[:FAIL] – did the latest measurement result value of a specific measurement fail?

Syntax	**:CALCulate:**{measProp}:LIMit[:FAIL]?
Returns	**0** when the limits were not violated or **1** in case at least one limit was violated by the latest measurement result value.
Description	Checks whether measurement result values of the **measurement specified with the {measProp} command element** are within their limits. In case, this command is used during a continuous measurement, only the **latest** measurement result will be checked. When this command is used subsequent to an array measurement, **all** measurement result values of the array will be checked against the limits. This means that a **1** will be returned if a single measurement result of an array is off the limits.

Example	**:MEAS:GSM:ARR:RFTX:LENG 10** **:MEAS:GSM:ARR:RFTX:PRMS 10** **:CALC:GSM:RFTX:LENG:LIM:FAIL?** In this example, the **:CALC** command will check whether all 10 measurement result values of the **LENG** measurement array are within their limits.

FAIL:CUMulative? – did any measurement result value fail?

Syntax	**CALCulate:LIMit:**FAIL:CUMulative?
Returns	**0** when the limits were not violated or **1** in case at least one limit was violated by at least one measurement result value.
Description	Checks whether **all** measurement result values of the measurement started last are within their limits.
Example	**:MEAS:GSM:ARR:RFTX:LENG 10** **:MEAS:GSM:ARR:RFTX:PRMS 10** **:CALC:GSM:LIM:FAIL:CUM?** This command of the CALCulate subsystem will check whether all measurement result values of the **PRMS** measurement are within their limits.

FAIL:CUM:RESet – resets the cumulative limit evaluation

Syntax	**CALCulate:LIMit:**FAIL:CUMulative:RESet
Description	Resets (clears) the cumulative check of measurement result values against their limits. Only the measurement results from **:MEAS:...** commands issued subsequent to this reset command will be taken into account for any limit checks.

{measProp}:LIMit:STATe – switches the limit check for a specific measurement on or off

Syntax	**CALCulate:**{measProp}:LIMit:STATe
Description	Switches the limit check for a measurement specified with the {measProp} command element either **ON** or **OFF**.
Example	**:CALC:RFTX:ALL:LIM:STAT OFF** Switches the limit check for the 19 main RF TX measurements **off**.

:{measProp}:LIMit:UPPer[:DATA] – sets the upper limit for a specific measurement

Syntax	**CALCulate:**{measProp}:LIMit:UPPer[:DATA]
Description	Sets the upper limit for the limit check of the measurement specified with the {measProp}; command element. The upper limit is the maximum measurement result allowed. Any measurement result value exceeding the value set with this command will result in a violation of the upper limit.

Example	**`:CALC:GSM:RFTX:PRMS:LIM:UPP 10.0`** This command sets the maximum RMS-valued phase error allowed to **`10.0`**. Any measurement result value exceeding this limit (e.g. 10.01) will result in a violation of the limits of the RMS-valued phase error measurement.

`:{measProp}:LIMit:LOWer[:DATA]` – sets the lower limit for a specific measurement

Syntax	**`CALCulate:`**`{measProp}:LIMit:LOWer[:DATA]`
Description	Sets the lower limit for the limit check of the measurement specified with the {measProp}; command element. The lower limit is the minimum measurement result allowed. Any measurement result value falling below the value set with this command will result in a violation of the lower limit.
Example	**`:CALC:AFAN:SIN:LIM:LOW 25.5`** This command sets the minimum SINAD required in order to pass the test to **`25.5`**. Any measurement result value falling below this limit (e.g. 25.4) will result in a violation of the limits of the RMS-valued phase error measurement.

CALCulate:{Statistics}

On the results of the last measurement performed, statistic test evaluation can be used. The related commands are described in this section.

NOTE

The commands described in this section will be reset with any new **`MEASure`** command. Therefore, the commands described below will only deliver the statistic evaluation of the measurement started last.

Syntax	`CALCulate:{Statistics}?`
Parameters	**`{Statistics}`** is one command element out of the following list: **`MAVerage	MMAXimum	MMINimum`**
Returns	A floating point real number representing the result of the related statistic evaluation, performed on all available results of the measurement started last.
Description	Calculates and returns the specified statistical property referring to the most recent measurement.
Examples	**`MEASure:GSM:ARRay:RFTX:ALL 10`** **`MEASure:GSM:ARRay:RFTX:PPEAk 10`** **`CALCulate:GSM:MMINimum?`** The first command takes 10 measurements of all RFTX parameters (190 then all together); the second one takes ten measurements of the peak phase error. The **`CALC:MMIN?`** command will only deliver back the minimum result of the peak phase error measurement as this was the measurement started last. The value delivered back in the result string is **`"5.05"`** in this example.

Using limits

The question 'do the performance characteristics of a mobile stay within the limits set by the appropriate specifications' is the core question of all GSM testing.
The **`LIMit`** subsystem of the 3100 offers a broad range of:

- defining single or complex limits, using the **[:DATA]** command element of the **CALCulate** subsystem and
- reading out the results of the limit checks, using the **:LIMit[:FAIL]?** query.

The limit evaluation system can be switched on or off for every single measurement parameter using the **:LIMit:SATe** command element.

[:DATa] - customize or query limits

Syntax	CALCulate<{measProp}>:LIMit<{limType}>[:DATa] <Val>
Parameters	**limType** is a placeholder for **:UPPer** or **:LOWer**. **Val** is the numeric value (floating) for the limit of the specific measurement parameter.
Description	Sets or queries the limit value(s) for the limit identified with **limType**. The limit evaluation will check the measurement results obtained against the limits set, using this command. **Note:** Some measurement types have more complex limits (such as the power/time template or the modulation spectrum). Please find a detailled explanation on those complex limits in subsection "Working with complex limits" on page 47.
Examples	**:CALCulate:GSM:RFTX:PRMS:LIMit:UPPer:DATa 4.00** Sets the upper limit of the RMS phase error to **4.00**. **:CALC:GSM:RFTX:PPEA:LIM:UPP?** Queries the currently set upper result limit for the peak phase error. The value will be returend as floating in the result string like **6.35**.

[:FAIL]? - pass/fail result query

Syntax	CALCulate<{measProp}>:LIMit[:FAIL]?
Returns	A boolean value or array (dependent on the type of measurement as defined by **measProp**). When all measurements of a type are within the limits, a **0** (pass) will be returned. if at least one mesaurement result is beyond the limits, a **1** (fail) will be returend.
Description	Checks whether any of the current measurement(s) failed to meet the limits. The type of measurement is defined by **measProp**.
Examples	**MEASure:GSM:ARRay:RFTX:POWer 10** Starts a series of 10 measurements of the RF peak power. **CALCulate:GSM:RFTX:POWer:LIMit:FAIL?** This command reads out the result of the 10 measurements checked against the limit. When all 10 measurement results are within the limits, a single **0** will be returned. If one or more measurement results are beyond the limits, a single **1** will be returned.

:STATe - switch limit evaluation on/off

Syntax	CALCulate<{measProp}>:LIMit:STATe <limState>
Parameters	**limState** is either **ON** or **OFF**.

Description	Activates or deactivates the check of the measurement results against the limit of the measurement type defined by **measProp**. **Note:** When the limit evaulation has been switched off, a **:FAIL** query will only return **0**(s) (pass).
Example	**CALCulate:GSM:RFTX:PRMS:LIMit:STATe OFF** This command switches the limit evaluation of the RMS phase error **off**. **MEASure:GSM:RFTX:PRMS** Starts a series of RMS phase error measurements. **CALCulate:GSM:RFTX:PRMS:LIMit:FAIL?** This query of the pass/fail evaluation will always return **0** (pass) as the limit evaluation of this parameter has been switched off.

Working with complex limits

It is not possible to define all relevant limits by just one number. Some limits are quite complex, like the power/time template, the corner points or the modulation spectrum.

Limits for the power/time template

The limits for the power/time template are made up by a total of 16 vectors; 9 for the upper limits and 7 for the lower limits.
These vectors have the following format: x,y.

- where **x** is the time in microseconds in relation to the beginning of the burst (i.e. the first bit of the useful part)
- and **y** is the RF power level in dB(c) in relation to the nominal output power level of the mobile.

Examples:

```
CALC:RFTX:TEMP:LIM:UPP -42,-47,-28,-47,-18,-28,-10,
-4,0,4, 552.8,1,560.8,-4,570.8,-28,580,-47
```

```
CALC:RFTX:TEMP:LIM:LOW 0,-150,0,-150,0,-40,20,
-1,270,-1,543,-1,543,-150
```

Limits for the corner points

The corner points are a maximum of eight positions to check critical parts of the burst. For each position, a minumum and a maximum RF power level may be specified.

- Positions are specified in microseconds in relation to the beginning of the burst (i.e. the first bit of the useful part)
- minimum and maximum RF power levels are specified in dB(c) in relation to the nominal output power level of the mobile.

Examples:

```
CALC:RFTX:CORN:POS -28,-18,-10,0,542.8,
552.8,560.8,570.8
```

```
CALC:RFTX:CORN:LIM:LOW -150,-150,-150,-150,
-150,-150,-150,-150
```

```
CALC:RFTX:CORN:LIM:UPP 4,4,4,4,4,4,4,4
```

Limits for the modulation spectrum

In case of the modulation spectrum, 23 positions (i.e. frequencies) have been predefined (see subsection Generator/Analyser for details).
With the related commands, the upper and lower limits for those 23 positions may be set. Those limits are specified in dB.

Measurement device configuration subsystems

These subsystems provide commands for setting and reading out the states of the various measurement devices of the 3100.
The following subsystems are available and described in this subsection.

RF measurement devices

RFGenerator – The radio frequency generator provides both continuous signals and bursts according to the specifications of the system (e.g. GSM) currently set. The RFGenerator subsystem controls the accessible parameters of the RF generator, such as RF level, frequency etc.

RFANalyser – This subsystem gives access to the setup parameters of the RF Analyser, such as frequency or trigger mode.

RFSPectrum – The commands of this subsystem are used to set span and resolution of the RF spectrum analyser.

AF measurement devices

AFGenerator – The audio frequency generator provides signals for audio measurements.
The AFGenerator subsystem controls the accessible parameters of the AF generator, such as output level, frequency, multitone signals etc.

AFANalyzer – The AF analyzer measures audio parameters. The related subsystem gives access to the settings.

MS Power Supply – The MS power supply option is an (optional) external device that simulates the power supply (battery) of a mobile under test. Together with the 3100, it provides a lot of interesting performance data of the mobile station. The related subsystem gives access to all relevant settings.

The RFGenerator subsystem

The RFG subsystem controls the accessible parameters of the RF generator.

Important notes:

– The RF generator can only be used if all communication systems have been switched off (and unloaded) before.

- The RF generator fuctionality of the 3100 will enable you to even provide a base channel to allow the mobile under test to synchronize to the base station. However, as long as the RF generator is active, there will be **no call setup** and **no reaction to signaling**.
  Some of the data transmitted by the 3100 in the base channel can be set or altered using the SCPI commands described in section "The CONFigure subsystem" on page 30.
- A good way to set up the 3100 as a RF generator for **circuit-switched standard GSM** signals is the use of the **`:CONFigure:CSYStem GCGenana`** command.
  While working in this mode, a base channel will **not** be provided.
- To set up the 3100 as a RF generator for **both packet-switched and circuit-switched GSM** signals, use the **`:CONFigure:CSYStem GPGenana`** command instead.
  In this mode, a base channel can be provided (see command **`:RFG:GSM:MOD:CHAN`** for details).

The RFANalyser subsystem

The RFAN subsystem controls the accessible parameters of the RF analyzer.

The RFSPectrum subsystem

The RFSP subsystem controls the accessible parameters of the RF modulation spectrum analyzer.

The AFGenerator subsystem

The AFG subsystem controls the accessible parameters of the audio generator. Please note that all commands of this subsystem require the Audio Option to be installed on your 3100.

The AFANalyser subsystem

The AFAN subsystem controls the accessible parameters of the AF analyzer. Please note that all commands of this subsystem require the Audio Option to be installed on your 3100.

The MS Power Supply subsystem

The PSUPply subsystem controls the accessible parameters of the MS Power Supply Option.

SCPI command errors

This subsection contains a table of SCPI command errors.

NOTE

If a query for an error code returns "0" then no error did occur.

Error Code	Meaning
61	QNX semaphore error. The operating system of the 3100 encountered a flag communication error.
62	TMSG QNX send error.
63	TMSG QNX sync error.
64	Internal communication error.
65	Unknown message received.
66	GPIB cannot be initialized.
67	This command is invalid.
68	Internal error of the task state system.
69	Error of the GPIB system.
70	QNX proxy error.
71	Process coordination error.
72	Message sent to the GPIB system is not understood there.
75	Error within the error message system.
78	Language expression invalid.
80	Name attachment error.
81	Proxy attachment error.
82	Proxy detach error.
83	Timer attachment error.
85	Timer delete error.
86	Parameter can not be set.
87	INI file error.
88	The file selected was not found.
89	DSP setup error.
100	Command error.
101	Invalid character.
102	Syntax error.
103	Invalid separator.
104	Data type error.
108	Parameter not allowed.
109	Parameter missing.
111	Header separator error.
112	Program mnemonic too long.
113	Undefined header.

114	Header suffix out of range.
121	Invalid character within a number.
123	Exponent too large.
128	Numeric data not allowed in this context.
131	Invalid suffix.
134	Suffix too long.
138	No suffix allowed in this context.
141	Invalid character data.
144	Character data too long.
158	No string allowed in this context.
160	Block data error.
168	No block data allowed in this context.
200	General execution error.
201	Multislot not active.
202	The external synchronization frequency is not within the ranges specified.
203	External synchronization changed during remote operation.
204	The operation is not possible in the current state of the 3100.
221	Settings of the 3100 lead to a conflict.
222	Data out of range.
225	No communication system running.
226	Timeout occurred while waiting for an uplink message to arrive from the mobile.
227	Layers 2/3 failed. Communication could not be established due to problems on the layer 2 and/or layer 3 level.
228	Mobile can not work in the enhanced frequency range.
229	No call release while an SMS is in progress.
230	Generator/Analyzer not running.
231	System running – no system expected.
250	Mass storage error.
253	Corrupt media.
256	File name not found.
272	Macro execution error.
280	Program error.
310	System error.

320	Save/recall memory lost.
330	Function not supported.
350	Queue overflow.
362	There needs to be an active call in order to start the codec option.
364	No audio hardware. The audio option would be required to complete a command, but it is not installed.
365	No codec hardware. The codec option would be required to complete a command, but it is not installed.
370	No results available.
371	Fetch: timeout occurred.
372	Fetch: no BER synchronization.
373	Fetch: arb data.
374	Measurement task error.
375	Error in burst data encountered.
376	ACPM receive error.
377	Autotemplate error.
378	Setting value for modulation spectrum out of range.
399	Invalid error code.
401	General CDMA measurement error: Invalid return code for sample acquisition.
402	CDMA Measurement error: Signal level too high for current input attenuation.
403	CDMA Measurement error: Signal level dropped below valid level during acquisition.
407	CDMA Measurement error: Sample acquisition RAM failure.
408	CDMA Measurement error: Signal level below minimum accuracy specification.
411	CDMA Measurement error: DSP EEPROM error. Default correction data loaded, measurement accuracy not verified.
413	CDMA Measurement error: synchronization to mobile signal failed.
415	CDMA Measurement error: Cannot trigger on mobile signal.

SCPI Command Reference

2

This SCPI Command Reference describes the commands for test automation using RAPID!, the General Purpose Interface Bus (GPIB), a USB, an RS-232 or a TCP/IP connection. The SCPI commands are divided into the following subsystems:

Signaling operation register group

This group of registers is 16 bits wide. Its main task is to deal with events related to the signaling status. The signaling operations depend on the system option. The Result bit of this group of registers is forwarded to bit 8 of the general operation register group.

Bit	Decimal	Meaning
0	1	The 3100 is in idle mode.
1	2	The 3100 is paging the mobile.
2	4	A call has been established and is currently active.
3	8	The BER loop has been closed.
4	16	Async mode (e.g. Multislot Generator/Analyzer) active.
5	32	The 3100 is calling the mobile under test. A BS call is in progress.
6	64	Not used (reserved).
7	128	Not used (reserved).
8	256	Alerting.
9	512	Not used (reserved).
10	1024	Not used (reserved).
11	2048	Not used (reserved).
12	4096	Not used (reserved).
13	8192	Not used (reserved).
14	16384	Not used (reserved).
15	32768	Not used (reserved).

Important Notes

Following are important notes regarding specific SCPI commands or general usage conditions.

NOTE:

The commands listed in the following sections are valid from 3100 firmware version 11.10. Especially WCDMA BER measurements are only supported from this version number onwards.

NOTE:

When you are using a USB or RS-232 interface, every command, which is sent to the 3100 will be answered by a CR LF (Carriabe Return Line Feed) command.

Common commands

The common commands are understood by all SCPI and IEEE 488.2 instruments. Their purpose is to perform general tasks and to read or work with the registers common to all SCPI instruments. The following common commands have been implemented.

General common commands

*CLS

Syntax	`:*CLS`
Parameters	There are no parameters.
Description	Resets the entire status reporting system: – The service register will be cleared (all bits will be set to **0**). – The event status register will be cleared (all bits will be set to **0**). – The error message queue will be emptied. – All event-type registers will be cleared.
Query	There is no query form of this command available.

*IDN?

Syntax	`:*IDN?`
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	Returns a string, containing the following information: – manufacturer's name – name of the device – serial number – software revision number All parameters are separated by commas. **Note:** In times of company mergers and acquisitions, it is a good idea to check the name of the device, not the manufacturer's name which may change between software updates. This does not preclude any name changes at Willtek but rather applies to instrumentation in general.
Example	`"WILLTEK, 3100, 0511099, 3.10.0001"`

***OPC**

Syntax	:*OPC
Parameters	There are no parameters.
Description	Postpones the execution of a command until all commands issued previously have been completed.
Query	Returns the 'operation complete' flag in a string. A **1** indicates that all commands have been completed while a **0** means that there is at least one command still under execution.

***RST?**

Syntax	:*RST?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	Resets the entire test set. All parameters, limits etc. will be set to the internally pre-defined default values.

***WAI**

Syntax	:*WAI
Parameters	There are no parameters.
Description	Postpones the execution of a command until all commands issued previously have been completed.
Query	There is no query form of this command available.

Commands affecting the event status register

The event status register contains eight bits. The meaning of these bits is outlined in the table below.
The commands working on the event status register are described below the table.

Bit	Decimal	Meaning
7	128	Power on – this bit is always set.
6	64	User Request – a **1** on this position indicates that the 3100 is no longer controlled by remote commands but by user interaction.
5	32	Command error – this bit indicates that one of the SCPI command errors occurred.
4	16	Execution error – is set after a SCPI execution error did occur.

3	8	Device-dependent error – this bit indicates that a device-specific SCPI error did occur.
2	4	Query error – is set after a SCPI query error occurred.
1	2	Request control – this bit is reserved for future use.
0	1	Operation complete flag – is set as soon as the execution of a command has been completed.

***ESE**

Syntax	`:*ESE <int1>`
Parameters	**int1** is an integer. The valid range is from 0 to 255 (8 bits).
Description	Sets the enable filter (mask) of the event status register. **int1** is the decimal representation of the binary mask. The mask and the current content of the event status register will be ANDed. If the result is not zero, then bit 5 of the Service register will be set.
Query	The query form reads out the enable filter (mask) currently set and returns its binary representation in a string.
Example	`:*ESE 128` As soon as power has been switched on, bit 7 (Power on) will be set. ANDed with the mask **128**, a binary **1** will occur and thus bit 5 of the service register will be set.

***ESR?**

Syntax	`:*ESR?`
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	Returns the decimal representation of the current contents of the event status register in a string. **Note:** This register is self-destructive, i.e. its contents will be cleared after reading.
Example	After power-on, an `:*ESR?` command will return **"128"**. This means that bit 7 is set and all the other bits of the event status register are **0**. The command will clear the event status register and a subsequent `:*ESR?` command will return **"0"**.

Commands affecting the service register

The service register represents the highest level within the report structure of the 3100.
The service register contains eight bits.
If any one of the bits 0...5 or 7 is set, the summary status bit (bit 6) of the service register will be set as well.

NOTE

The service register is self-destructive. This means that its contents will be cleared after reading.

Bit	Decimal	Meaning
7	128	OPERational status summary. When this bit is set, an event within the general operation register group (e.g. the 3100 is waiting for a trigger) passed all filters.
6	64	Summary status bit. This bit will always be set as soon as any other bit of the service register has been set. **Note:** The summary status bit may be ANDed with the service request enable filter in order to generate a **S**ervice Re**Q**uest on GPIB. The related command is **`*SRE`** (see below for details).
5	32	Event status summary. When this bit is set, an event within the event status register group (e.g. an error occurred) passed all filters.
4	16	Message available. This bit will be set to **1** as soon as a query has been completed and measurement result(s) are available.
3	8	QUESTionable status summary. If this bit is set, an event within the general questionable status register group (e.g. 'value out of range') passed all filters.
2	4	Error queue status. When this bit is set, the error queue contains error messages. Up to 10 error messages can be logged in the error queue. The error queue can be read out, using the :SYSTem:ERRor? command.
1	2	Remote command completed. This bit will be set to **1** after a remote (SCPI) command has been completed. **Note:** However, when the 3100 receives a SCPI command, it will block the GPIB until the command has been completed.
0	1	Message queue status. This bit will be set to **1** as soon as a message is available in the 3100's internal message queue. Up to 10 messages can be logged in the message queue. To write to or to read from the message queue, use the :SYSTem:MESSage command.

*SRE

Syntax	:*SRE <int1>
Parameters	**int1** is an integer. The valid range is from 0 to 255 (8 bits).
Description	Sets the enable filter (mask) for the service register. **int1** is the decimal representation of this binary mask. The mask and the current content of the service register will be ANDed. If the result is not zero, a service request (SRQ) will occur on the GPIB.
Query	The query form reads out the mask currently set and returns its binary representation in a string.
Example	**:*SRE 68** As soon as an error occurs, bits 2 and 6 of the service register will be set. ANDed with the mask (**68**), a binary **1** will be the result and a SRQ will occur on the GPIB.

*STB?

Syntax	:*STB?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	Returns the decimal representation of the current contents of the service register in a string. **Note:** This register is self-destructive, i.e. its contents will be cleared after reading.
Example	A **:*STB?** command returns **"68"**. The return value of **68** (= **64** + **4**) means that an error occurred (**4**).

SYSTem subsystem

System parameters, such as the number of unread error messages, the test set's GPIB address etc. may be read out or set using the commands of this subsystem.

:SYSTem:ERRor[:NEXT]?

Syntax	:SYSTem:ERRor[:NEXT]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the oldest unread error message from the internal error queue of the Willtek 3100. The queue entry returned will be a string containing the error no. and additional text. The maximum length of the string is 255 characters.
Example	**:SYSTem:ERRor:NEXT?** String returned: **"66 GPIB cannot be initialised."**

:SYSTem:ERRor:COUNt?

Syntax	:SYSTem:ERRor:COUNt?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the number of unread error messages in the internal error queue of the Willtek 3100. The string returned will contain one integer. The maximum number of errors stored internally is 10.
Example	**:SYSTem:ERRor:COUNt?** String returned: **"0"** This means that there are no unread error messages in the error queue.

:SYSTem:ERRor:CODE[:NEXT]?

Syntax	:SYSTem:ERRor:CODE[:NEXT]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the code of the oldest unread error message in the internal error queue of the Willtek 3100. The string returned will contain one integer (and no text).
Example	**:SYSTem:ERRor:CODE?** String returned: **"66"** This means that the GPIB could not be initialized.

:SYSTem:ERRor:CODE:ALL?

Syntax	:SYSTem:ERRor:CODE:ALL?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the error codes of all unread error messages in the internal error queue of the Willtek 3100. The string returned will contain a maximum of 10 integers, separated by commas.
Example	**:SYSTem:ERRor:CODE:ALL?** String returned: **"371,66"** This means that there were two unread error messages in the error queue (the first one indicating that there was a timeout on a **FETCh** command and the second one meaning that the GPIB could not be initialized).

:SYSTem:ERRor:REMote:DISPlay

Syntax	:SYSTem:ERRor:REMote:DISPlay <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: OFF|ON. Default is OFF.
Description	Switches the error display on the start screen on and off. If it is switched to ON, the error log will be displayed after the first error occurred.
Query	Reads and returns the current setting.
Example	**:SYST:ERR:REM:DISP ON** In remote case the error log window will be displayed.

:SYSTem:Message

Syntax	:SYSTem:Message <string1>
Parameters	**string1** is a string (text) parameter. The maximum length of **string1** is 255 characters.
Description	Writes the message specified with the **string1** parameter to the Willtek 3100's internal system **message** queue.
Query	Reads and returns the oldest unread message in the Willtek 3100's internal **message** queue. The string returned will contain a maximum of 255 characters.
Example	**:SYSTem:MESSage "23.17,Procedure A5"** **:SYST:MESS?** String returned: **"23.17,Procedure A5"** In this example, a RAPID! program performs some internal calculations and then writes the result to the system message queue. This result is then read by the external controller.

:SYSTem:COMMunicate:LOCal

Syntax	:SYSTem:COMMunicate:LOCal
Parameters	There are no parameters.
Description	Sets up the Willtek 3100 to allow manual operation on the front panel during SCPI operation. **Note:** This command may be used e.g. to allow interactive alignment procedures in a production or quality assurance flow.
Query	There is no query form of this command available.
Example	:SYSTem:COMM:LOC

:SYSTem:COMMunicate:GPIB:ADDRess

Syntax	:SYSTem:COMMunicate:GPIB:ADDRess <int1>[,<int2>]
Parameters	**intx** are two integers. The minimum value for **int1** is **1**, the maximum is **32**. The default value is **4**. The minimum value for **int2** is **0**, the maximum is **30**. The default value is **1**. **int1 must** be specified while **int2** is an optional parameter.
Description	Sets the GPIB address of the Willtek 3100. For details regarding the GPIB address, refer to section Setting the GPIB address.
Query	Reads and returns the current setting of the GPIB address as explained above.
Example	**:SYST:COMM:GPIB:ADDR 14** Sets the GPIB address to 14.

:SYSTem:COMMunicate:GPIB:TERMinator

Syntax	:SYSTem:COMMunicate:GPIB:TERMinator <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **LF|CR|CRLF**. Default is **LF**.
Description	Sets the terminator used on the GPIB. For details regarding the terminator, refer to section Setup.
Query	Reads and returns the current setting of the terminator used on the GPIB as explained above.
Example	**:SYST:COMM:GPIB:TERM CRLF** Sets the GPIB terminator to 'carriage return plus line feed'.

:SYSTem:COMMunicate:TCPip:ADDRess

Syntax	:SYSTem:COMMunicate:TCPip:ADDRess <string>
Parameters	**string** is a string only containing the IP address for the 3100.
Description	This command sets the IP address of the 3100. See section I/O Configuration for more details.
Query	Reads and returns the current setting of the IP address as explained above.
Example	**SYST:COMM:TCP:ADDR "192.16.16.114"** sets the IP address to a defined value.

:SYSTem:COMMunicate:TCPip:NETMask

Syntax	:SYSTem:COMMunicate:TCPip:NETMask <string>
Parameters	**string** is a string only containing the net mask for the 3100.
Description	This command sets the net mask of the 3100. See section I/O Configuration for more details.
Query	Reads and returns the current setting of the net mask as explained above.
Example	**SYST:COMM:TCP:NETM "255.255.255.0"** sets the net mask to a defined value.

:SYSTem:COMMunicate:TCPip:GATeway

Syntax	:SYSTem:COMMunicate:TCPip:GATeway <string>
Parameters	**string** is a string only containing the default gateway address for the 3100.
Description	This command sets the default gateway address of the 3100. See section I/O Configuration for more details.
Query	Reads and returns the current setting of the gateway address as explained above.
Example	**SYST:COMM:TCP:GAT "192.16.16.1"** sets the gateway address to a defined value.

:SYSTem:COMMunicate:TCPip:PORT

Syntax	:SYSTem:COMMunicate:TCPip:PORT <int>
Parameters	**int** defines the TCP/IP port address of the 3100. The address must be in the range from 49152 to 65535.
Description	This command sets the port address on which the 3100 can be controlled via LAN. See section I/O Configuration for more details.
Query	Reads and returns the current setting of the port used by TCPIP as explained above.
Example	**SYST:COMM:TCP:PORT 49200** sets the TCP/IP port address to its default.

:SYSTem:COMMunicate:TCPip:TERMinator

Syntax	:SYSTem:COMMunicate:TCPip:TERMinator <PredefExp>
Parameters	**PredefExp** can take on one of the following values: **LF** or **CR** or **CRLF**. The default is **LF**.
Description	The command defines the delimiter for SCPI control strings. See section I/O Configuration for more details.
Query	Reads and returns the current setting of the terminator used by TCPIP as explained above.
Example	**SYST:COMM:TCP:TERM CRLF** sets the line terminator for SCPI commands via LAN to CR (Carriage Return) followed by LF (Line Feed).

:SYSTem:COMMunicate:TCPip:MOUNt

Syntax	:SYSTem:COMMunicate:TCPip:MOUNt <string1> <string2>
Parameters	**string1** defines the network address which shall be mounted as a device for remote control. The maximum allowable length of the string is 255 characters. **string2** is the symbolic device address used in SCPI programming. The maximum allowable length of **string2** is 25 characters. The default is "server".
Description	This command mounts a server disk as a 3100 device which can be used to load or save data to/from. See section I/O Configuration for more details.
Query	Reads and returns the last settings for the mount path and the corresponding local name as explained above.
Example	:SYST:COMM:TCP:MOUNt "unixpc/disk2/results","resdir"

:SYSTem:COMMunicate:TCPip:DHCP

Syntax	:SYSTem:COMMunicate:TCPip:DHCP <PredefExp>
Parameters	PredefExp can take on one of the following values: ON or OFF. The default is OFF.
Description	The command turns DHCP on or off. See section I/O Configuration for more details.
Query	Reads and returns the current setting of DHCP operation.
Example	:SYST:COMM:TCP:DHCP ON sets the software to use DHCP.

:SYSTem:COMMunicate:TCPip:DHCP

Syntax	:SYSTem:COMMunicate:TCPip:DHCP <PredefExp>
Parameters	PredefExp can take on one of the following values: ON or OFF. The default is OFF.
Description	The command turns DHCP on or off. See section I/O Configuration for more details.
Query	Reads and returns the current setting of DHCP operation.
Example	:SYST:COMM:TCP:DHCP ON sets the software to use DHCP.

:SYSTem:COMMunicate:TCPip:WCDMa:ADDRess

Syntax	`:SYSTem:COMMunicate:TCPip:WCDMa:ADDRess <string>`
Parameters	`string` is a string only containing the IP address for the 3100 WCDMA board.
Description	This command sets the IP address of the 3100 WCDMA board.
Query	Reads and returns the current setting of the IP address as explained above.
Example	`SYST:COMM:TCP:WCDMA:ADDR "192.16.16.114"` sets the IP address to a defined value.

:SYSTem:COMMunicate:TCPip:WCDMa:NETMask

Syntax	`:SYSTem:COMMunicate:TCPip:WCDMa:NETMask <string>`
Parameters	`string` is a string only containing the net mask for the 3100 WCDMA board.
Description	This command sets the Netmask of the 3100 WCDMA board.
Query	Reads and returns the current setting of the IP address as explained above.
Example	`SYST:COMM:TCP:WCDMA:NETM "255.255.255.0"` sets the Netmask to a defined value.

:SYSTem:COMMunicate:TCPip:WCDMa:GATeway

Syntax	`:SYSTem:COMMunicate:TCPip:WCDMa:GATeway <string>`
Parameters	`string` is a string only containing the default gateway address for the 3100 WCDMA board.
Description	This command sets the default gateway address of the 3100 WCDMA board.
Query	Reads and returns the current setting of the IP address as explained above.
Example	`SYST:COMM:TCP:WCDMA:GAT "192.16.16.1"` sets the Gateway address to a defined value.

:SYSTem:COMMunicate:TCPip:WCDMa:HOSTaddress

Syntax	`:SYSTem:COMMunicate:TCPip:WCDMa:HOSTaddress <string>`
Parameters	`string` is a string only containing the Host IP address for the 3100 WCDMA board.
Description	This command sets the Host IP address of the 3100 WCDMA board.
Query	Reads and returns the current setting of the Host IP address as explained above.
Example	`SYST:COMM:TCP:WCDMA:HOST "192.16.16.114"` sets the Host IP address to a defined value.

:SYSTem:COMMunicate:SERA:PARameter

Syntax	:SYSTem:COMMunicate:SERA:PARameter <int1>,<int2>,<int3>,<PredefExpr4>
Parameters	There are four parameters. **int1** is the bit rate on the serial interface. Valid values are **110,300,600,1200,2400,4800,9600,19200,38400,57600, 115200**. The default value is **38400**. **int2** is the number of bits per character. The minimum value is **5**, the maximum is **8**. The default value is **8**. **int3** is the number of stop bits. It can take on the values **1** or **2**. The default value is **1**. **<PredefExpr4>** specifies the parity bit. The value is one of the following predefined expressions: **NO | ODD | EVEN**. Default is **NO**.
Description	Sets the parameters for serial port COM1. This command uses the following format: **baud,bits,stop,parity** where **baud** stands for the bit rate (**int1**), **bits** stands for the number of bits per character (**int2**) , **stop** stands for the number of stop bits (**int3**) and **parity** represents the parity (No, Odd or Even)(**PredefExpr4**).
Query	Reads and returns the current settings of COM1 as explained above.
Example	**:SYST:COMM:SERA:PAR 9600,8,1,ODD** **:SYST:COMM:SERA:PAR?** String returned: **"9600,8,1,ODD"**.

:SYSTem:COMMunicate:SERA:BAUD

Syntax	:SYSTem:COMMunicate:SERA:BAUD <int1>
Parameters	int1 is the bit rate on the serial interface. Valid values are 9600,19200,38400,57600,115200. The default value is 57600.
Description	Sets the baud rate for serial port COM1. The other serial parameter settings are fixed at 8 bits per char, 1 stop bit, no parity.
Query	Reads and returns the current settings of COM1 as explained above.
Example	:SYST:COMM:SERA:BAUD 115200 :SYST:COMM:SERA:BAUD? Returns the following string: "9600".

:SYSTem:KEYBoard

Syntax	:SYSTem:KEYBoard <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **USA|BELGium_fr|BELGium_nl|CANFr|CANEng|DEN|FR|GER|ITA|JAP|LATAm_spa|LATAm_port |NL|NOR|PORTugal|SPA|SWE|SWISs_fr |SWISs_ger|UK**. Default is **USA**.
Description	Selects the language for the external keyboard. **Note:** The language setting defines the position and type of special characters.
Query	Reads and returns the current setting of the language of the external keyboard. The string returned will contain one of the predefined expressions as explained above.
Example	**:SYST:KEYB SWE** Sets the language for the external keyboard to 'Swedish'.

:SYSTem:DATE

Syntax	:SYSTem:DATE <int1>,<int2>,<int3>
Parameters	**intx** are three integers. The minimum value for **int1** is **1998**, the maximum is **2100**. The default value is **1998**. The minimum value for **int2** is **1**, the maximum is **12**. The default value is **1**. The minimum value for **int3** is **1**, the maximum is **31**. The default value is **1**.
Description	Sets the system date. This command uses the following format: **jjjjmmdd** where **jjjj** stands for the four digits of the year (**int1**), **mm** gives the two digits of the current month (**int2**) and, **dd** represents the day of the current month (**int3**).
Query	Reads and returns the current system date in a string, using the format explained above.
Example	**:SYST:DATE 2001,7,6** Sets the system date to the 6th of July, 2001.

:SYSTem:TIME

Syntax	:SYSTem:TIME <int1>,<int2>,<int3>
Parameters	**intx** are three integers. The minimum value for **int1** is **0**, the maximum is **23**. The default value is **0**. The minimum value for **int2** is **0**, the maximum is **59**. The default value is **0**. The minimum value for **int3** is **0**, the maximum is **59**. The default value is **0**.
Description	Sets the system time. This command uses the following format: **hhmmss** where **hh** stands for the two digits of the current hour, using a 24 hour time format (**int1**), **mm** gives the two digits of the current minute (**int2**) and, **ss** represents the seconds of the system time (**int3**).
Query	Reads and returns the current system time in a string, using the format explained above.
Example	**:SYST:TIME?** String returned: **"12,56,05"**.

:SYSTem:VERSion?

Syntax	:SYSTem:VERSion?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	Reads and returns the version number of the SCPI command system used on your Willtek 3100. The command will return a string, containing one floating point real number.
Example	**:SYST:VERS?** String returned: **"2001.7"**.

:SYSTem:PRINTer

Syntax	`:SYSTem:PRINTer <PredefExp>`
Parameters	**PredefExp** is one of the following predefined expressions: **HP400|HP680|EPST|HPLJ|EPSP|EPSX|EPSI|EPCI|EPS1|EPPX|EPC2|EPC4|EPC5|EPC6|EPC8|EPC1|EPC3|BMPF|CANO**. Default is **HP400**.
Description	Selects the printer for screen dumps. The printers selectable are: **HP400** means the Hewlett-Packard (HP) deskjet 400 series, **HP680** stands for the DP deskjet 680 series, **EPST** means the Epson (EP) stylus series, **HPLJ** stands for the HP laserjet series, **EPSP**, **EPSX**, **EPSI**, **EPCI**, **EPS1**, **EPPX**, **EPC1**, **EPC2**, **EPC3**, **EPC4**, **EPC5**, **EPC6**, and **EPC8** stand for the corresponding Epson printer series, **BMPF** means that the screen dump will be output as a bit map file, while **CANO** will generate an output signal suitable for Canon printers.
Query	Reads and returns the current setting for the printer. The string returned will contain one of the predefined expressions as explained above.
Example	**:SYST:PRIN?** String returned: **"HPLJ"**.

STATus subsystem

The STATus subsystem offers commands to read out and deal with

- the general operation register group and its subordinate groups of registers and
- the general questionable status register group and its subordinate groups of registers.

NOTE

The commands to deal with the event status register group and the service register are part of the SCPI command errors.

`:STATus:PRESet`

Syntax	`:STATus:PRESet`
Parameters	There are no parameters.
Description	This command sets all user-definable settings of the status subsystem to their factory default values. The default values for the single commands are explained below.
Query	There is no query form of this command available.
Example	**`:STATus:PRESet`** Will reset all parameters of the status subsystem to their default values as listed below.

`:STATus:GSM:SUMMary`

Syntax	`:STATus:GSM:SUMMary`
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	The query form of this command returns a summary of important status registers. The string returned contains an integer, representing a 16 bit word. **Note:** Bit 15 (the MSB) is not used. Therefore, the maximum value returned will be **`"32767"`**.
Example	`:STAT:GSM:SUMM?`

:STATus:OPERation[:EVENt]?

Syntax	:STATus:OPERation[:EVENt]?
Parameters	There are no parameters.
Description	Reads out the current contents of the General Operation Event Register. **Note:** Event-type registers are read-only and self-destructive. They will be cleared after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The value returned represents all general operation events that have occurred since the last query of this register. As with any event-type register, the single bits will remain set even when the reason for the bits to be set is no longer valid. Please note that this is the main difference between event-type and condition-type registers. Condition-type registers reflect the current state of the Willtek 3100. Consequently, condition-type registers will be updated continuously. **Notes** – In case, a certain event shall be trapped in a loop within a program, always query the event-type register.
Example	**:STATus:OPERation:EVENt?** Value returned: "32". This means that bit 5 (the 'waiting for a trigger' bit) has been set. This indicates that a 'waiting for a trigger' event did occur. If you want to know whether the Willtek 3100 is still waiting for the trigger to occur, read out the related condition-type register (see command explained below).

:STATus:OPERation:CONDition?

Syntax	:STATus:OPERation:CONDition?
Parameters	There are no parameters.
Description	This command reads out the current contents of the General Operation Condition Register. This register reflects the current operational state of the Willtek 3100 and will be updated continuously. **Note:** This register is nondestructive. This means that it will keep its contents after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The integer will express all bits currently set in the 16 bit general operation condition register. **Note:** Bit 15 (the MSB) of this register is not used. Therefore, the maximum value returned will be **"32767"**.
Example	**:STATus:OPERation:CONDition?** Value returned: "512". This means that bit 9 (the MEASure summary bit) has been set, indicating that some measurement is currently in progress. Bit 9 will be reset as soon as the measurement has been completed.

:STATus:OPERation:ENABLe

Syntax	:STATus:OPERation:ENABLe <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	This command sets the Enable filter of the general operation register group. This mask will be **AND**ed with the general operation **event** register and thus decide what kind of events will be forwarded to bit 7 of the service register. **Note:** The service register is often also referred to as the statusbyte register or status byte.
Query	There is no query form of this command available.
Example	**:STATus:OPERation:ENABle 129** This means that any setting of bits 1 (calibrating) **or** 7 (correcting) of the general operation event register will rise bit 7 of the service register.

:STATus:OPERation:NTRansition

Syntax	:STATus:OPERation:NTRansition <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	Sets the NTR mask of the general operation register group. This mask will be **AND**ed with the up to 15 bits of the NTR transition filter of the general operation **condition** register to allow a reset (negative transition) of any bit (i.e. a transition from logic '1' to '0') to reach the general operation **event** register. **Notes** – The default of this mask is 0 – that means that the mask will **not** allow any negative transition of the lower 15 bits of the condition-type register to reach the event-type register. – Bit 16 of the general operation condition register is not used.
Query	There is no query form of this command available.
Example	**:STATus:OPERation:NTRansition 32767** This means that all of the negative transitions of the condition-type register will be forwarded to the event-type register.

:STATus:OPERation:PTRansition

Syntax	:STATus:OPERation:PTRansition <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **32767**.
Description	Sets the PTR mask of the general operation register group. This mask will be **AND**ed with the up to 15 bits of the PTR transition filter of the general operation **condition** register to allow a positive transition of any bit (i.e. a transition from logic '0' to '1') to reach the general operation **event** register. Notes – The default of this mask is 32767 – that means that the mask will allow any positive transition of the lower 15 bits of the condition-type register to reach the event-type register. – Bit 16 of the general operation condition register is not used.
Query	There is no query form of this command available.
Example	**:STATus:OPERation:PTRansition 0** This means that none of the positive transitions of the condition-type register will be forwarded to the event-type register.

:STATus:OPERation:SIGNalling:GSM[:EVENt]?

Syntax	:STATus:OPERation:SIGNalling:GSM[:EVENt]?
Parameters	There are no parameters.
Description	Reads out the current contents of the signaling operation event register. **Note:** Event-type registers are read-only and self-destructive. They will be cleared after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The value returned represents all signaling operation events that have occurred since the last query of this register. As with any event-type register, the single bits will remain set even when the reason for the bits to be set is no longer valid. Please note that this is the main difference between event-type and condition-type registers. Condition-type registers reflect the current state of the Willtek 3100. Consequently, condition-type registers will be updated continuously. **Note:** In case, a certain event shall be trapped in a loop within a program, always query the event-type register.
Example	**:STATus:OPERation:SIGNalling:GSM:EVENt?** Value returned: "8". This means that bit 3 of the signaling operation event register (stands for BS-initiated call) has been set. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STAT:OPERation:SIGNalling:GSM:CONDition?

Syntax	:STAT:OPERation:SIGNalling:GSM:CONDition?
Parameters	There are no parameters.
Description	This command reads out the current contents of the signaling operation condition register. This register reflects the current state of the Willtek 3100 in terms of signaling and will be updated continuously. **Note:** This register is nondestructive. This means that it will keep its contents after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The integer will express all bits set in the 16 bit signaling operation condition register. **Note:** Bits 14 and 15 (the MSB) of this register are not used. Therefore, the maximum value returned will be **"16383"**.
Example	**:STATus:OPERation:SIGNalling:GSM:CONDition?** Value returned: "512". This means that bit 9 has been set and that there is a change of the call channel currently in progress. Bit 9 will be reset as soon as the change of the call channel has been completed. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STATus:OPERation:SIGNalling:GSM:ENABLe

Syntax	:STATus:OPERation:SIGNalling:GSM:ENABLe <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	This command sets the enable filter of the signaling operation register group. This mask will be **AND**ed with the signaling operation **event** register and thus decide what kind of events will be forwarded to bit 8 of the general operation register group.
Query	There is no query form of this command available.
Example	**:STATus:OPERation:SIGNalling:GSM:ENABLe 16** This means that any setting of bit 4 (MS clearing) of the signaling operation event register will rise bit 8 of the general operation condition register. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STAT:OPERation:SIGNalling:GSM:PTRansition

Syntax	`:STAT:OPERation:SIGNalling:GSM:PTRansition <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **32767**.
Description	Sets the PTR mask of the Signaling operation register group. This mask will be **AND**ed with the up to 15 bits of the PTR filter of the signaling operation **condition** register to allow a positive transition of any bit (i.e. a transition from logic '0' to '1') to reach the signaling operation **event** register. **Note:** The default of this mask is 32767 – that means that the mask will allow any positive transition of the lower 15 bits of the condition-type register to reach the event-type register.
Query	There is no query form of this command available.
Example	**:STATus:OPERation:SIGNalling:GSM:PTRansition 0** This means that none of the positive transitions of the condition-type register will be forwarded to the event-type register. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STAT:OPERation:SIGNalling:GSM:NTRansition

Syntax	`:STAT:OPERation:SIGNalling:GSM:NTRansition <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	Sets the NTR mask of the Signaling operation register group. This mask will be **AND**ed with the up to 15 bits of the NTR filter of the signaling operation **condition** register to allow a reset (negative transition) of any bit (i.e. a transition from logic '1' to '0') to reach the signaling operation **event** register. **Note:** The default of this mask is 0 – that means that the mask will **not** allow any negative transition of the lower 15 bits of the condition-type register to reach the event-type register of this group.
Query	There is no query form of this command available.
Example	**:STATus:OPERation:SIGNalling:GSM:NTRansition 0** This means that none of the negative transitions of the condition-type register will be forwarded to the event-type register. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STATus:OPERation:SIGNalling:WCDMa[:EVENt]?

Syntax	`:STATus:OPERation:SIGNalling:WCDMa[:EVENt]?`
Parameters	There are no parameters.
Description	Reads out the current contents of the signaling operation event register. **Note:** Event-type registers are read-only and self-destructive. They will be cleared after any query.
Query	There is only a query form of this command available. The query will return a string containing one integer. The value returned represents all signaling operation events that have occurred since the last query of this register. As with any event-type register, the individual bits will remain set even when the reason for the bits to be set is no longer valid. Please note that this is the main difference between event-type and condition-type registers. Condition-type registers reflect the current state of Willtek 3100. Consequently, condition-type registers will be updated continuously. **Note:** In case, a certain event shall be trapped in a loop within a program, always query the event-type register.
Example	`:STATus:OPERation:SIGNalling:WCDMa:EVENt?` Value returned in this example: "8". This means that bit 3 of the signaling operation event register has been set. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STAT:OPERation:SIGNalling:WCDMa:CONDition?

Syntax	`:STAT:OPERation:SIGNalling:WCDMa:CONDition?`
Parameters	There are no parameters.
Description	This command reads out the current contents of the signaling operation **condition** register. This register reflects the current state of Willtek 3100 in terms of signaling and will be updated continuously. **Note:** This register is non-destructive. This means that it will keep its contents after any query.
Query	There is only a query form of this command available. The query will return a string containing one integer. The integer will express all bits set in the 16 bit signaling operation condition register. **Note:** For further details regarding the basic functions of the STATus subsystem, please refer to section Understanding the STATus Subsystem.
Example	`:STATus:OPERation:SIGNalling:WCDMa:CONDition?` Value returned in this example: "512". This means that bit 9 has been set. Bit 9 will be reset as soon as the change of the call channel has been completed. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STATus:OPERation:SIGNalling:WCDMa:ENABLe

Syntax	`:STATus:OPERation:SIGNalling:WCDMa:ENABLe <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is `0`, the maximum value is `32767`. The value default is `0`.
Description	This command sets the enable filter of the signaling operation register group. This mask will be **AND**ed with the signaling operation **event** register and thus decide what kind of events will be forwarded to bit 8 of the general operation register group.
Query	There is no query form of this command available.
Example	`:STATus:OPERation:SIGNalling:WCDMa:ENABle 16` This means that any setting of bit 4 of the signaling operation event register will rise bit 8 of the general operation condition register. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STAT:OPERation:SIGNalling:WCDMa:PTRansition

Syntax	`:STAT:OPERation:SIGNalling:WCDMa:PTRansition <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is `0`, the maximum value is `32767`. The value default is `32767`.
Description	Sets the PTR mask of the signaling operation register group. This mask will be ANDed with the up to 15 bits of the PTR filter of the signaling operation condition register to allow a positive transition of any bit (i.e. a transition from logic '0' to '1') to reach the signaling operation event register. **Note** The default of this mask is `32767` - that means that the mask will allow any positive transition of the lower 15 bits of the condition-type register to reach the event-type register.
Query	There is no query form of this command available.
Example	`:STATus:OPERation:SIGNalling:WCDMa:PTRansition 0` This means that none of the positive transitions of the condition-type register will be forwarded to the event-type register. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STAT:OPERation:SIGNalling:WCDMa:NTRansition

Syntax	`:STAT:OPERation:SIGNalling:WCDMa:NTRansition <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is `0`, the maximum value is `32767`. The value default is `0`.
Description	Sets the NTR mask of the signaling operation register group. This mask will be **AND**ed with the up to 15 bits of the NTR filter of the signaling operation **condition** register to allow a reset (negative transition) of any bit (i.e. a transition from logic '1' to '0') to reach the signaling operation **event** register. **Note:** The default of this mask is 0 - that means that the mask will **not** allow any negative transition of the lower 15 bits of the condition-type register to reach the event-type register of this group.
Query	There is no query form of this command available.
Example	`:STATus:OPERation:SIGNalling:WCDMa:NTRansition 0` This means that none of the negative transitions of the condition-type register will be forwarded to the event-type register. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 54.

:STATus:OPERation:MEASuring[:EVENt]?

Syntax	`:STATus:OPERation:MEASuring[:EVENt]?`
Parameters	There are no parameters.
Description	Reads out the current contents of the measuring operation event register. **Note:** Event-type registers are read-only and self-destructive. They will be cleared after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The value returned represents all measuring operation events that have occurred since the last query of this register. As with any event-type register, the single bits will remain set even when the reason for the bits to be set is no longer valid. Please note that this is the main difference between event-type and condition-type registers. Condition-type registers reflect the current state of the Willtek 3100. Consequently, the condition type registers will be updated continuously. **Notes** In case, a certain event shall be trapped in a loop within a program, always query the event-type register.
Example	**`:STATus:OPERation:MEASuring:EVENt?`** Value returned: "`8`". This means that bit 3 (stands for AF measurement) has been set.

:STAT:OPERation:MEASuring:CONDition?

Syntax	:STAT:OPERation:MEASuring:CONDition?
Parameters	There are no parameters.
Description	This command reads out the current contents of the measuring operation condition register. This register reflects the current state of the Willtek 3100 in terms of measurements and will be updated continuously. **Note:** This register is nondestructive. This means that it will keep its contents after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The integer will express all bits set in the 16 bit measuring operation condition register. **Note:** Bits 4 to 15 (the MSB) of this register are not used. Therefore, the maximum value returned will be **"7"**.
Example	**:STATus:OPERation:MEASuring:CONDition?** Value returned: "1". This means that bit 0 has been set and that there is an RFTX measurement currently in progress. This bit will be reset as soon as the RFTX measurements have been completed.

:STATus:OPERation:MEASuring:ENABLe

Syntax	:STATus:OPERation:MEASuring:ENABLe <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	This command sets the enable filter of the measuring operation register group. This mask will be **AND**ed with the measuring operation **event** register and thus decide what kind of events will be forwarded to bit 9 of the general operation register group.
Query	There is no query form of this command available.
Example	**:STATus:OPERation:MEASuring:ENABle 4** This means that any setting of bit 2 (RF spectrum) of the measuring operation event register will rise bit 9 of the general operation condition register.

:STAT:OPERation:MEASuring:PTRansition

Syntax	:STAT:OPERation:MEASuring:PTRansition <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **32767**.
Description	Sets the PTR mask of the measuring operation register group. This mask will be **AND**ed with the up to 16 bits of the PTR filter of the measuring operation **condition** register to allow a positive transition of any bit (i.e. a transition from logic '0' to '1') to reach the measuring operation **event** register. **Note:** The default of this mask is 32767 – that means that the mask will allow any positive transition of the lower 15 bits of the condition-type register to reach the event-type register.
Query	There is no query form of this command available.
Example	**:STATus:OPERation:MEASuring:PTRansition 0** This means that none of the positive transitions of the condition-type register will be forwarded to the event-type register.

:STAT:OPERation:MEASuring:NTRansition

Syntax	:STAT:OPERation:MEASuring:NTRansition <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	Sets the NTR mask of the measuring operation register group. This mask will be **AND**ed with the up to 16 bits of the NTR filter of the measuring operation **condition** register to allow a reset (negative transition) of any bit (i.e. a transition from logic '1' to '0') to reach the measuring operation **event** register. **Note:** The default of this mask is 0 – that means that the mask will **not** allow any negative transition of the 16 bits of the condition-type register to reach the event-type register of this group.
Query	There is no query form of this command available.
Example	**:STATus:OPERation:MEASuring:NTRansition 0** This means that none of the negative transitions of the condition-type register will be forwarded to the event-type register.

:STATus:QUEStionable[:EVENt]?

Syntax	:STATus:QUEStionable[:EVENt]?
Parameters	There are no parameters.
Description	Reads out the current contents of the general questionable status event register. **Note:** Event-type registers are read-only and self-destructive. They will be cleared after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The value returned represents all general questionable status events that have occurred since the last query of this register. As with any event-type register, the single bits will remain set even when the reason for the bits to be set is no longer valid. Please note that this is the main difference between event-type and condition-type registers. Condition-type registers reflect the current state of the Willtek 3100. Consequently, condition-type registers will be updated continuously. **Note:** In case, a certain event shall be trapped in a loop within a program, always query the event-type register.
Example	**:STATus:QUEStionable:EVENt?** Value returned: "256". This means that bit 8 has been set, indicating that the calibration of the Willtek 3100 is out of range. If you want to know whether the Willtek 3100 calibration is still out of range, read out the related condition-type register (see command explained below).

:STATus:QUEStionable:CONDition?

Syntax	:STATus:QUEStionable:CONDition?
Parameters	There are no parameters.
Description	This command reads out the current contents of the general questionable status condition register. This register reflects the current questionable state of the Willtek 3100 and will be updated continuously. **Note:** This register is nondestructive. This means that it will keep its contents after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The integer will express all bits currently set in the 16 bit questionable status condition register. **Notes**: Bit 15 (the MSB) of this register is not used. Therefore, the maximum value returned will be **"32767"**.
Example	**:STATus:QUEStionable:CONDition?** Value returned: "512". This means that bit 9 (the RF summary bit) has been set, indicating a current problem on the RF side of the Willtek 3100.

:STATus:QUEStionable:ENABLe

Syntax	`:STATus:QUEStionable:ENABLe <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	This command sets the Enable filter of the general questionable status register group. This mask will be **AND**ed with the general questionable status **event** register and thus decide what kind of events will be forwarded to bit 3 of the service register. **Note:** The service register is often also referred to as the statusbyte register or status byte.
Query	There is no query form of this command available.
Example	**:STATus:QUEStionable:ENABle 512** This means that any setting of bit 9 (RF summary bit) of the general questionable status event register will rise bit 3 of the service register.

:STATus:QUEStionable:PTRansition

Syntax	`:STATus:QUEStionable:PTRansition <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **32767**.
Description	Sets the PTR mask of the general questionable status register group. This mask will be **AND**ed with the up to 16 bits of the PTR filter of the general questionable status **condition** register to allow a positive transition of any bit (i.e. a transition from logic '0' to '1') to reach the questionable status **event** register. **Note:** The default of this mask is 32767 – that means that the mask will allow any positive transition of the lower 15 bits of the condition-type register to reach the event-type register.
Query	There is no query form of this command available.
Example	**:STATus:QUEStionable:PTRansition 0** This means that none of the positive transitions of the condition-type register will be forwarded to the event-type register of this group.

:STATus:QUEStionable:NTRansition

Syntax	`:STATus:QUEStionable:NTRansition <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	Sets the NTR mask of the general questionable status register group. This mask will be **AND**ed with the up to 16 bits of the NTR filter of the general questionable status **condition** register to allow a reset (negative transition) of any bit (i.e. a transition from logic '1' to '0') to reach the general questionable status **event** register. **Note:** The default of this mask is 0 – that means that the mask will **not** allow any negative transition of the 16 bits of the condition-type register to reach the event-type register.
Query	There is no query form of this command available.
Example	**:STATus:QUEStionable:NTRansition 0** This means that none of the negative transitions of the condition-type register will be forwarded to the event-type register of this group.

:STATus:QUEStionable:RF[:EVENt]?

Syntax	`:STATus:QUEStionable:RF[:EVENt]?`
Parameters	There are no parameters.
Description	Reads out the current contents of the RF questionable status event register. **Note:** Event-type registers are read-only and self-destructive. They will be cleared after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The value returned represents all RF questionable status events that have occurred since the last query of this register. As with any event-type register, the single bits will remain set even when the reason for the bits to be set is no longer valid. Please note that this is the main difference between event-type and condition-type registers. Condition-type registers reflect the current state of the Willtek 3100. Consequently, condition-type registers will be updated continuously. **Note:** In case, a certain event shall be trapped in a loop within a program, always query the event-type register.
Example	**:STATus:QUEStionable:RF:EVENt?** Value returned: "1". This means that bit 1 has been set, indicating an RF input overload. If you want to know whether this RF input overload still persists, read out the related condition-type register (see command explained below).

:STATus:QUEStionable:RF:CONDition?

Syntax	:STATus:QUEStionable:RF:CONDition?
Parameters	There are no parameters.
Description	This command reads out the current contents of the RF questionable status condition register. This register reflects the current questionable state of the Willtek 3100 regarding RF and will be updated continuously. **Note:** This register is nondestructive. This means that it will keep its contents after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The integer will express all bits currently set in the 16 bit RF questionable status condition register. **Note:** Bits 4 to 15 (the MSB) of this register are not used. Therefore, the maximum value returned will be **"15"**
Example	**:STATus:QUEStionable:RF:CONDition?** Value returned: "8". This means that bit 3 has been set, indicating that the frequency currently received is out of range of the Willtek 3100 receiver.

:STATus:QUEStionable:RF:ENABLe

Syntax	:STATus:QUEStionable:RF:ENABLe <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	This command sets the enable filter of the RF questionable status register group. This mask will be **AND**ed with the RF questionable status **event** register and thus decide what kind of events will be forwarded to bit 9 of the general questionable status register group.
Query	There is no query form of this command available.
Example	**:STATus:QUEStionable:RF:ENABle 15** This means that any setting of one of the lower four bits of the RF questionable status event register will rise bit 9 of the general questionable status condition register.

:STATus:QUEStionable:RF:PTRansition

Syntax	`:STATus:QUEStionable:RF:PTRansition <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **32767**.
Description	Sets the PTR mask of the RF questionable status register group. This mask will be **AND**ed with the up to 16 bits of the PTR filter of the RF questionable status **condition** register to allow a positive transition of any bit (i.e. a transition from logic '0' to '1') to reach the RF questionable status **event** register. **Note:** The default of this mask is 32767 – that means that the mask will allow any positive transition of the lower 15 bits of the condition-type register to reach the event-type register of this group.
Query	There is no query form of this command available.
Example	**:STATus:QUEStionable:RF:PTRansition 0** This means that none of the positive transitions of the condition-type register will be forwarded to the event-type register.

:STATus:QUEStionable:RF:NTRansition

Syntax	`:STATus:QUEStionable:RF:NTRansition <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	Sets the NTR mask of the RF questionable status register group. This mask will be **AND**ed with the up to 16 bits of the NTR filter of the RF questionable status **condition** register to allow a negative transition of any bit (i.e. a transition from logic '1' to '0') to reach the RF questionable status **event** register. **Note:** The default of this mask is 0 – that means that the mask will **not** allow any negative transition of the 16 bits of the condition-type register to reach the event-type register of this group.
Query	There is no query form of this command available.
Example	**:STATus:QUEStionable:RF:NTRansition 0** This means that none of the negative transitions of the condition-type register will be forwarded to the event-type register.

:STATus:QUEStionable:SYNChron[:EVENt]?

Syntax	:STATus:QUEStionable:SYNChron[:EVENt]?
Parameters	There are no parameters.
Description	Reads out the current contents of the synchronization questionable status event register. **Note:** Event-type registers are read-only and self-destructive. They will be cleared after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The value returned represents all synchronization questionable status events that have occurred since the last query of this register. As with any event-type register, the single bits will remain set even when the reason for the bits to be set is no longer valid. Please note that this is the main difference between event-type and condition-type registers. Condition-type registers reflect the current state of the Willtek 3100. Consequently, condition-type registers will be updated continuously. **Note:** In case, a certain event shall be trapped in a loop within a program, always query the event-type register.
Example	**:STATus:QUEStionable:SYNChron:EVENt?** Value returned: "2". This means that bit 2 has been set, indicating that an external frame synchronization signal has been recognized on the SYNC IN/OUT socket on the rear panel of the Willtek 3100.

:STAT:QUEStionable:SYNChron:CONDition?

Syntax	:STAT:QUEStionable:SYNChron:CONDition?
Parameters	There are no parameters.
Description	This command reads out the current contents of the synchronization questionable status condition register. This register reflects the current questionable state of the Willtek 3100 regarding synchronization and will be updated continuously. **Note:** This register is nondestructive. This means that it will keep its contents after any query.
Query	There is only a query form of this command available. The query will return a string, containing one integer. The integer will express all bits currently set in the 16 bit synchronization questionable status condition register. **Note:** Bits 2 to 15 (the MSB) of this register are not used. Therefore, the maximum value returned will be **"3"**.
Example	**:STATus:QUEStionable:SYNChron:CONDition?** Value returned: "1". This means that bit 1 has been set, indicating that an external RF synchronization signal is being received on the EXT SYNC prog. socket on the real panel of the Willtek 3100.

:STATus:QUEStionable:SYNChron:ENABLe

Syntax	:STATus:QUEStionable:SYNChron:ENABLe <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	This command sets the Enable filter of the SYNChronization questionable status register group. This mask will be **AND**ed with the synchronization questionable status **event** register and thus decide what kind of events will be forwarded to bit 10 of the general questionable status register group.
Query	There is no query form of this command available.
Example	**:STATus:QUEStionable:SYNChron:ENABle 3** This means that any setting of one of the lower two bits of the synchronization questionable status event register will rise bit 10 of the general questionable status condition register.

:STAT:QUES:SYNChron:PTRansition

Syntax	:STAT:QUES:SYNChron:PTRansition <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **32767**.
Description	Sets the PTR mask of the SYNChronization questionable status register group. This mask will be **AND**ed with the up to 16 bits of the PTR filter of the synchronization questionable status **condition** register to allow a positive transition of any bit (i.e. a transition from logic '0' to '1') to reach the synchronization questionable status **event** register. **Note:** The default of this mask is 32767 – that means that the mask will allow any positive transition of the lower 15 bits of the condition-type register to reach the event-type register of this group.
Query	There is no query form of this command available.
Example	**:STAT:QUES:SYNChron:PTRansition 0** This means that none of the positive transitions of the synchronization questionable status condition register will be forwarded to the event-type register.

:STAT:QUES:SYNChron:NTRansition

Syntax	`:STAT:QUES:SYNChron:NTRansition <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **32767**. The default value is **0**.
Description	Sets the NTR mask of the SYNChronization questionable status register group. This mask will be **AND**ed with the up to 16 bits of the NTR filter of the synchronization questionable status **condition** register to allow a reset (negative transition) of any bit (i.e. a transition from logic '1' to '0') to reach the corresponding **event**-type register. **Note:** The default of this mask is 0 – that means that the mask will **not** allow any negative transition of the 16 bits of the condition-type register to reach the corresponding event-type register.
Query	There is no query form of this command available.
Example	**:STATus:QUEStionable:SYNChron:NTRansition 0** This means that none of the negative transitions of the condition-type register will be forwarded to the event-type register of this group.

PROGram subsystem

This subsystem contains commands related to loading and executing RAPID! program files.

:PROGram[:SELected]:NAME

Syntax	`:PROGram[:SELected]:NAME <string>`
Parameters	**string** is an existing RAPID! program file name. The maximum length of **string** is 50 characters.
Description	This command loads a RAPID! basic program file. To start or stop the program, please use the **:PROGram[:SELected]:STATus** command described below. The name of the RAPID! basic program file may contain a path.
Query	The query returns the name of the currently loaded RAPID! basic program file as a string.
Example	**:PROGram:SELected:NAME "test7389.bas"** **:PROG:NAME?** Value returned: **"test7389.bas"**.

:PROGram[:SELected]:NUMBer

Syntax	`:PROGram[:SELected]:NUMBer <string> [,int1] [,int2]`
Parameters	**string** is the name of an existing numeric variable in the currently loaded RAPID! program file. The maximum length of **string** is 100 characters. **int1** and **int2** are two integers. The minimum value for each **intx** is **-32765**, the maximum value is **32765**. The default value for each **intx** is **0**.
Description	This command exists in a query form only. It is used to read out the current value of a numeric variable of the currently loaded RAPID! basic program. The value will be returned as a floating point real number.
Query	The query returns the current value of a numerical variable as one floating point real number contained in a string.
Example	**:PROGram:SELected:NUMBer? "BER_Samples",1,0** This example will read out the current value of the variable **BER_Samples** at array position **1,0**. The result could be **20000.0**.

:PROGram[:SELected]:STRing

Syntax	`:PROGram[:SELected]:STRing <string> [,int1] [,int2]`
Parameters	**string** is the name of an existing string variable in the currently loaded RAPID! program file. The maximum length of **string** is 100 characters. **int1** and **int2** are two integers. The minimum value for each **intx** is **-32765**, the maximum value is **32765**. The default value for each **intx** is **0**.
Description	This command exists in a query form only. It is used to read out the current value of a string variable of the currently loaded RAPID! basic program. The value will be returned as a string.
Query	The query returns the current value of a string variable as a string.
Example	**:PROGram:SELected:STRing? "LST_Output_File$",2** This example will read out the current value of the variable **LST_output_file$** at array position (line) **2**. The result could be **"c:\3100\rapid\test7389.lst"**.

:PROGram[:SELected]:CRC

Syntax	`:PROGram[:SELected]:CRC`
Parameters	There are no parameters.
Description	This command exists in a query form only. It allows to read out the CRC Checksum of the RAPID! basic program currently loaded. The value will be returned as a long integer number. **Note:** A long integer number allows a maximum value of 2,147,483,647.
Query	The query returns the CRC checksum of the currently loaded RAPID! basic program as a string, containing one long integer number.
Example	**:PROGram:SELected:CRC?** The value returned could be **"5356362"**.

:PROGram[:SELected]:STATus

Syntax	`:PROGram[:SELected]:STATus <PredefExp>`
Parameters	**PredefExp** is one of the following predefined expressions: **RUN|STOP**. Default is **RUN**.
Description	Starts (**RUN**) or stops (**STOP**) the execution of the currently loaded RAPID! program file.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**:PROGram[:SELected]:STATus RUN** **:PROG:STAT?** Value returned: **"RUN"**.

:PROGram[:SELected]:MESSage

Syntax	:PROGram[:SELected]:MESSage <string>
Parameters	**string** is string with a maximum length of 255 characters.
Description	This command either reads (in its query form) or writes a string from/to the RAPID! program message queue.
Query	The query returns the latest entry of the RAPID! program message queue.
Example	**:PROGram:SELected:MESSage "Put this in the queue"** **:PROG:MESS?** The result read back will be **"Put this in the queue"**.

:PROGram[:SELected]:SRE

Syntax	:PROGram[:SELected]:SRE <int1>
Parameters	**int1** is an integer. Minimum value for **int1** is **0**, maximum value is **255**. Default value for **int1** is **0**.
Description	This command sets the mask for the service register of the RAPID! program. This is similar to the ***SRE** common command.
Query	There is no query form of this command available.
Example	**:PROGram:SELected:SRE 16** This will set bit 4 of the mask of the service register. The contents of the service register will be **AND**ed with this mask. As soon as there is a valid measurement result available for the measurement started last, bit 4 of the service register will be set to '1'. ANDed with this mask (bit 4 set to 1), a binary 1 will be the result and a service request will be issued.

:PROGram[:SELected]:STB

Syntax	:PROGram[:SELected]:STB
Parameters	There are no parameters.
Description	This command exists in a query form only. It reads out the contents of the service register of the RAPID! program. This is similar to the ***STB** common command.
Query	The command returns a string containing one integer. This integer represents the contents of the service register of the RAPID! program.
Example	**:PROGram:SELected:STB?** If a **4** is returned, this indicates that there is a valid measurement result available for the measurement started last.

FORMat subsystem

The FORMat subsystem sets and queries settings concerning the data output in remote mode.

:FORMat:MRESult:STYPe

Syntax	:FORMat:MRESult:STYPe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **STB|SIGNalling|MEASuring|OPERation |QUEStionable|ALL**. Default is that no additional information will be provided with the measurement result values.
Description	Specifies the type of additional information to be returned with any measurement result obtained by a FETCh command. The meaning of the settings is as follows: **STB** will deliver the current contents of the Service register. **SIGNalling** stands for the current contents of the signaling operation condition register. **MEASuring** means that the current contents of the measuring operation condition register will be delivered back. **OPERation** will deliver the current contents of the general operation condition register. **QUEStionable** stands for the current contents of the general questionable status condition register. **ALL** will deliver the current contents of the eight most important registers. The order is as follows: 1. Service register, 2. event status register, 3. general operation condition register, 4. signaling operation condition register, 5. measuring operation condition register, 6. general questionable status condition register, 7. RF questionable status condition register, and 8. synchronization questionable status condition register. The contents of every single register will be returned as an integer; the single values will be separated by commas.
Query	There is no query form of this command available.
Example	**:FORMat:MRESult:HEADer ON** **:FORMat:MRESult:STYPe ALL** **:MEASure:RFTX:PRMS** **:FETCh:RFTX:PRMS** The first command switches the transmission of the additional information on, the second command specifies that all current contents of the eight most important registers shall be returned. The third command starts a continuous measurement. The last command finally delivers the latest measurement result value plus the current contents of the main registers. The string delivered back: **"0,128,256,8,1,0,0,0,4.63"**.

:FORMat:MRESult:HEADer

Syntax	:FORMat:MRESult:HEADer <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **OFF|ON**. Default is **OFF**.
Description	Switches the transmission of the additional information (see explanation of the command above) either on or off. **ON** means that the current contents of the corresponding register(s) will be added **at the beginning** of every string returned by a FETCh command.
Query	There is no query form of this command available.
Example	**:FORM:MRES:HEAD** This command will switch the transmission of the additional information **off**.

:FORMat:ADELimiter

Syntax	:FORMat:ADELimiter <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **COMMa|COLOn|SEMIcolon**. Default is **COMMa**.
Description	Selects the delimiter to be used to separate parameters on commands **and** single measurement result values. **COMMa** stands for commas (default), **COLOn** sets the delimiter to be a colon (:), while **SEMIcolon** will use and expect a semicolon (;) to be used.
Query	There is no query form of this command available.
Example	**:FORM:ADEL** Defines the comma to be used as delimiter for both commands and measurement results.

:FORMat:RESolution

Syntax	:FORMat:RESolution <int1>
Parameters	**int1** is an integer. The minimum value for **<int1>** is **0**, the maximum is **20**. The default value is **6**.
Description	Defines the number of digits after the decimal point to be used for floating point real numbers.
Query	There is no query form of this command available.
Example	**:FORM:RES 0** Defines that there will be no digits after the decimal point.

CONFigure subsystem

This subsystem incorporates all changeable BS (GSM) and NB (WCDMA/UMTS) parameters of all systems implemented. The key commands are the following:

CONFigure:CSYStem	Selects the communications system to work with.
CONFigure:<SystemOption>:...	These commands select parameters within a communications system.
CONFigure:<SystemOption>:BS:... **CONFigure:<SystemOption>:NB:...**	These are the commands to set specific system parameters like the base station's RF output power level or its identity.
CONFigure:<SystemOption>:MSTAtion:... **CONFigure:<SystemOption>:UE:...**	The mobile-specific information is handed over to the 3100 using these commands. One example is the power level.
CONFigure:<SystemOption>:BER:...	These commands set the BER parameters.
CONFigure:<SystemOption>:GROup:...	With the help of these commands, groups of measurements may be defined.
CONFigure:COUPloss:...	These commands provide access to the coupling loss compensation feature of the 3100.

:CONFigure:CSYStem

Syntax	:CONFigure:CSYStem <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: NONe | GCGenana | GPGenana | EGPGenana | CDGenana | WCGenana | AMGenana | GSM | GPRS | EGPRs | CDMA | WCDMa. Default is NONe
Description	Selects the type of communication system to be used. NONe means that there is no system loaded and that there will be no basic generator or analyzer functionality available. This parameter may be used to cancel all RF radiation from the Willtek 3100. GENana will make the Willtek 3100 work as a generator and analyzer for circuit-switched GSM signals. This setting is identical to GCGenana. Because of the implementation of packet-data channels, we recommend not to use the GENana parameter in new programs but the GCGenana parameter as it clearly marks that the Willtek 3100 will generate and analyze circuit-switched GSM signals only with this setting. GCGenana will set up the Willtek 3100 as a generator and analyzer for circuit-switched GSM signals (including multislot/HSCSD). Generator/analyzer means that there will be no call setup and therefore, all measurements will be asynchronous. GPGenana will make the Willtek 3100 work as an asynchronous generator and analyzer for all kinds of GSM signals (circuit-switched signals in single or multislot mode as well as packet-data channels (PDTCH in GPRS)). GSM will set up the Willtek 3100 as a test set for circuit-switched GSM systems (including multislot/HSCSD). All tests performed with this setting require a call setup. Therefore, this test mode is called the 'call mode'. GPRS will bring the Willtek 3100 into call mode for standard GSM and GPRS systems. This means that this parameter will enable testing of all kinds of GSM signals (circuit-switched signals in single or multislot mode as well as packet-data channels (GPRS)). **Note:** Please keep in mind that you have to select the communication system first when working with SCPI, as the default of this command is NONe.
Query	The query form of this command will return the current setting. The string delivered back will contain the short-form version of one of the predefined expressions explained above.
Example	**:CONFigure:CSYStem GSM** **:CONF:CSYS?** Value returned: **"GSM"**.

:CONFigure:GSM:TYPE

Syntax	:CONFigure:GSM:TYPE <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **GSM9001800|GSM9001900**. Default is **GSM9001800**.
Description	Selects the type of GSM system to be used after the Willtek 3100 was brought into GSM call mode. **GSM9001800** will make the Willtek 3100 work as a test set for the GSM system types GSM 900 and GSM 1800. **GSM9001900** will set up the Willtek 3100 as a test set for the GSM system types GSM 900 and GSM 1900. Notes – This command will have no effect when the Willtek 3100 is in generator/analyzer mode. – The main reason for this command to exist is the 'double numbering' of channels. GSM 1800 and GSM 1900 use the same channel numbers above channel number 511, but the related physical channels are in different frequency bands.
Query	The query form of this command will return the current setting. The string delivered back will contain one of the predefined expressions as explained above.
Example	**:CONFigure:GSM:TYPE GSM9001900** **:CONF:GSM:TYPE?** Value returned: **"GSM9001900"**.

:CONFigure:GSM:MSLot

Syntax	:CONFigure:GSM:MSLot <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **OFF**.
Description	Turns multislot operation either on or off. **OFF** means that standard GSM is active and all multislot functions are disabled. **Note:** HSCSD or multislot testing requires the Multislot Option to be installed on your Willtek 3100.
Query	The query form of this command will return the current setting. The string delivered back will contain one of the predefined expressions as explained above.
Example	**:CONFigure:GSM:MSLot ON** **:CONF:GSM:MSL?** Value returned: **"ON"**.

:CONFigure:GSM:ASSAll

Syntax	:CONFigure:GSM:ASSAll <int1>,<int2>
Parameters	**int1** and **int2** are two integers. The minimum value for **int1** is **0**, the maximum value is **1023**. The default value for **int1** is **45**. The minimum value for **int2** is **0**, the maximum value is **31**. The default value for **int2** is **10**.
Description	This command changes the TCH (traffic channel) number and the mobile's power control level in one assignment. **int1** is the new TCH channel number and **int2** is the MS power control level. **Note:** The RF power control level is a code. The corresponding RF output power of the mobile under test depends on the type of GSM system and the frequency range.
Query	The query form of this command will return the current setting. The string delivered back will contain two integers. The first one expresses the TCH number and the second one the MS power control level.
Example	**:CONFigure:GSM:ASAll 45,10** **:CONF:GSM:ASA?** Value returned: **"45,10"**.

:CONFigure:GSM:BS:LEVel

Syntax	:CONFigure:GSM:BS:LEVel <real1>
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **-120.0**, the maximum value **-10.0**. The minimum resolution possible for **real1** is **0.1**. The default value for **real1** is **-60.0**.
Description	This command sets the RF output power level of the Willtek 3100 (i.e. the simulated base station). The physical dimension of **real1** is dBm.
Query	The query form of this command will return the current RF output power setting of the Willtek 3100. The string delivered back will contain one real number.
Example	**:CONFigure:GSM:BS:LEVel -50.5** **:CONF:GSM:BS:LEV?** Value returned: **"-50.5"**.

:CONFigure:GSM:BS:CMODe

Syntax	:CONFigure:GSM:BS:CMODe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **FACCh | SDCCh**.
Description	sets the mode of call setup to FACCH or SDCCH
Query	Returns the mode of call setup
Example	**:CONFigure:GSM:BS:CMODe FACCh** **:CONFigure:GSM:BS:CMODe?** Value returned: **"FACCh"**.

:CONFigure:GSM:BS:LAI:MCC

Syntax	:CONFigure:GSM:BS:LAI:MCC <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **1000**. The default value for **int1** is **1**.
Description	This command sets the mobile country code. More details regarding the MCC can be found in section Basic System Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:LAI:MCC 238** **:CONF:GSM:BS:LAI:MCC?** Value returned: **"238"**.

:CONFigure:GSM:BS:LAI:MNC[:DATA]

Syntax	:CONFigure:GSM:BS:LAI:MNC[:DATA] <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **99** for a two-digit MNC and **999** for a three-digit MNC; see command :CONF:GSM:BS:LAI:MNC:FORMat. The default value for **int1** is **1**.
Description	This command sets the mobile network code. More details regarding the MNC can be found in section Basic System Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:LAI:MNC 20** **:CONF:GSM:BS:LAI:MNC?** Value returned: **"20"**.

:CONFigure:GSM:BS:LAI:MNC:FORM

Syntax	:CONFigure:GSM:BS:LAI:MNC:FORM <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **TWODigits|THREedigits**. Default is **TWODigits**.
Description	This command sets the format of the mobile network code.
Query	The query form of this command will return the current setting. The string delivered back will contain one of the predefined expressions as explained above.
Example	**:CONFigure:GSM:BS:LAI:MNC:FORM THRE** **:CONF:GSM:BS:LAI:MNC:FORM?** Value returned: **"THRE"**.

:CONFigure:GSM:BS:LAI:LAC

Syntax	:CONFigure:GSM:BS:LAI:LAC <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **65535**. The default value for **int1** is **1**.
Description	This command sets the location area code. For details, please refer to section Basic System Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:LAI:LAC 17399** **:CONF:GSM:BS:LAI:LAC?** Value returned: **"17399"**.

:CONFigure:GSM:BS:NCC

Syntax	:CONFigure:GSM:BS:NCC <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **7**. The default value for **int1** is **2**.
Description	This command sets the network color code. For details, please refer to section Basic System Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:NCC 2** **:CONF:GSM:BS:NCC?** Value returned: **"2"**.

:CONFigure:GSM:BS:BCC

Syntax	:CONFigure:GSM:BS:BCC <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **7**. The default value for **int1** is **0**.
Description	This command sets the base station color code. For details, please refer to section Basic System Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:BCC 1** **:CONF:GSM:BS:BCC?** Value returned: **"1"**.

:CONFigure:GSM:BS:PUIT

Syntax	:CONFigure:GSM:BS:PUIT <int1>
Parameters	**int1** is an integer. The minimum value for int1 is **0**, the maximum value is **255**. The default value for **int1** is **0**.
Description	This command sets the periodic updating interval timer T3212. For details, please refer to section Basic System Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:PUIT 1** **:CONF:GSM:BS:PUIT?** Value returned in this example: "**1**".

:CONFigure:GSM:BS:BCH:ARFCn

Syntax	:CONFigure:GSM:BS:BCH:ARFCn <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **1023**. The default value for **int1** is **63**.
Description	This command sets the channel number for the Willtek 3100 base channel (BCCH). For details, please refer to section Test Parameters. **Note:** The base channel is sometimes also referred to as the 'channel number of the main carrier'.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:BCH:ARFCn 60** **:CONF:GSM:BS:BCH:ARFC?** Value returned: **"60"**.

:CONFigure:GSM:BS:TCH:ARFCn

Syntax	:CONFigure:GSM:BS:TCH:ARFCn <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **1023**. The default value for **int1** is **45**.
Description	This command sets the traffic channel (TCH) number to be used. For details, please refer to section Test Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:TCH:ARFCn 917** **:CONF:GSM:BS:TCH:ARFC?** Value returned: **"917"**.

:CONFigure:GSM:BS:TCH:TSLot

Syntax	:CONFigure:GSM:BS:TCH:TSLot <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **2**, the maximum value is **6**. The default value for **int1** is **2**.
Description	This command sets the time slot to be used for the traffic channel (TCH). For details, please refer to section Basic System Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:TCH:TSLot 5** **:CONF:GSM:BS:TCH:TSL?** Value returned: **"5"**.

:CONFigure:GSM:BS:TCH:TYPE

Syntax	:CONFigure:GSM:BS:TCH:TYPE <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **FR|EFR**. Default is **FR**.
Description	This command sets the type of traffic channel. **FR** stands for full rate and **EFR** for enhanced full rate. For details, please refer to section Test Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one of the predefined expressions as explained above.
Example	**:CONFigure:GSM:BS:TCH:TYPE EFR** **:CONF:GSM:BS:TCH:TYPE?** Value returned: **"EFR"**.

:CONFigure:GSM:BS:NCEL

Syntax	:CONFigure:GSM:BS:NCELl [int1],[int2],[int3],[int4],[int5],[int6]
Parameters	**int1** to **int6** are integers. The minimum value for each **intx** is **0**, the maximum value is **1023**. The default value for each **intx** is **0**.
Description	This command is used to set the channel numbers for up to six neighboring cells. These channel numbers are broadcast in the system information message by the Willtek 3100. The mobile under test has to perform a MS report on those frequency channels. For details, please check with section MS Report. **Notes** – All six integer values are optional. If the command is sent without any channel number specified, the default values (=0) will be broadcast by the Willtek 3100 in the system information message. – Let's assume that there was a first command sent with six channel numbers specified. After that, a second command is sent with only two channel numbers specified. Now, the Willtek 3100 will alter only the first two channel numbers and maintain all other settings (see example below for details).

Query	The query form of this command will always return the complete set of neighboring channel numbers currently in use. The string delivered back will always contain six integers.
Example	**:CONFigure:GSM:BS:NCELl 10,20,30,40,50,60** **:CONF:GSM:BS:NCEL 70,80** **:CONF:GSM:BS:NCEL?** Values returned: **"70,80,30,40,50,60"**.

:CONFigure:GSM:BS:CI

Syntax	:CONFigure:GSM:BS:CI <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **255**. The default value for **int1** is **255**.
Description	This command sets the cell identity parameter (i.e. the radio cell number), broadcast in the system information message. For more details on the CI, please refer to section Basic System Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:CI 127** **:CONF:GSM:BS:CI?** Value returned: **"127"**.

:CONFigure:GSM:BS:CBA

Syntax	:CONFigure:GSM:BS:CBA <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **1**. The default value for **int1** is **0**.
Description	This command sets the cell barred access parameter. A **0** set for the CBA will allow the mobile to contact the base station, while a **1** will block the base station and the mobile under test will not be granted access.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:CBA 1** **:CONF:GSM:BS:CBA?** Value returned: **"1"**.

:CONFigure:GSM:BS:MSLot[:DATA]

Syntax	:CONFigure:GSM:BS:MSLot[:DATA] <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **1**, the maximum value is **4**. The default value for **int1** is **2**.
Description	This command sets the number of time slots to be transmitted by the Willtek 3100 during multislot operation. **Note:** HSCSD or multislot testing requires the Multislot Option to be installed on your Willtek 3100.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BS:MSLot:DATA 4** **:CONF:GSM:BS:MSL?** Value returned: **"4"**. This means that the Willtek 3100 will transmit multislot data during 4 time slots.

:CONFigure:GSM:BS:MSLot:LEVel[:DATA]

Syntax	:CONFigure:GSM:BS:MSLot:LEVel[:DATA] <real1>,<real2>,<real3>,<real4>
Parameters	**real1...real4** are four floating point real numbers. The minimum value for each **realx** is **-120.0**, the maximum value is **-10.0**. The minimum resolution of each **realx** is **0.1**. The default value for each **realx** is **-60.0**. **Note:** All four values **must** be specified.
Description	This command configures individual RF power levels for the up to four DL time slots during Multislot-HSCSD operation. Every value represents the RF output power level of the Willtek 3100 in dBm during the related time slot. **Notes** – Please note that this command only prepares the individual RF power levels. To activate them, use the **:CONFigure:GSM:BS:MSLot:LEVel:MODE** command described below. – This command requires the Multislot Option to be installed on the Willtek 3100.
Query	The query form of this command will return the current setting. The string delivered back will always contain four real numbers.
Example	**:CONFigure:GSM:BS:MSLot:LEVel:DATA -80.0,-20.5,-60.0, -60.0** **:CONF:GSM:BS:MSL:LEV:DATA?** Values returned: **"-80.0,-20.5,-60.0,-60.0"**

:CONFigure:GSM:BS:MSLot:LEVel:MODE

Syntax	:CONFigure:GSM:BS:MSLot:LEVel:MODE <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **INDLev|STDLev**. Default is **INDLev**.
Description	Switches the use of individual levels for multislot operation on or off. **INDLev** means that the individual levels, specified with the **:CONF:GSM:BS:MSL:LEV** command will be used. **STDLev** means that the individual RF power levels will be ignored and that the standard RF output level of the Willtek 3100 (as set with the **:CONF:GSM:BS:LEV** command) will be used.
Query	The query form of this command will return the current setting. The string delivered back will contain the short-form version of one of the predefined expressions as explained above.
Example	**:CONFigure:GSM:BS:MSL:LEV:MODE INDLev** **:CONF:GSM:BS:MSL:LEV:MODE?** Value returned: **"INDL"**.

:CONFigure:GSM:BS:ATTach

Syntax	:CONFigure:GSM:BS:ATTach <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **OFF**.
Description	Switches the IMSI Attach/Detach flag in the System Information messages broadcast on the BCCH on or off. If ON, the mobile must register and deregister with the network simulated by the Willtek 3100 when switched on or off, respectively. This can be used to detect when the mobile is switched on and listening to the Willtek 3100 network.
Query	The query form of this command will return the current setting. The string delivered back will contain one of the predefined expressions as explained above.
Example	**:CONFigure:GSM:BS:ATTach OFF** **:CONF:GSM:BS:ATT?** Value returned: **"OFF"**.

:CONFigure:GSM:MSTAtion:DRX

Syntax	:CONFigure:GSM:MSTAtion:DRX <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **7**. The default value for **int1** is **0**.
Description	This command sets the BS-PA-MFRMS parameter, broadcast in the control channel description information element. This parameter is used for discontinuous reception (DRX). For more information about DRX, please refer to section Basic System Parameters. **Note:** The value entered here is coded according to the following scheme: the number of multiframes used is the number entered here plus 2. This means that the default value (**0**) will lead to a DRX period of two multiframes.

Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:MSTAtion:DRX 2** **:CONF:GSM:MSTA:DRX?** Value returned: **"2"**, resulting in a DRX period of four multiframes.

:CONFigure:GSM:MSTAtion:TADVance

Syntax	:CONFigure:GSM:MSTAtion:TADVance <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **63**. The default value for **int1** is **0**.
Description	This command sets the timing advance for the mobile under test to use. The current setting of the timing advance is broadcast to the mobile in the SACCH physical header. **Note:** More details regarding the timing advance can be found in section Time Slots and Bursts.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:MSTAtion:TADVance 12** **:CONF:GSM:MSTA:TADV?** Value returned: **"12"**.

:CONFigure:GSM:MSTAtion:PLEVel[:ALL]

Syntax	:CONFigure:GSM:MSTAtion:PLEVel[:ALL] <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **31**. The default value for **int1** is **10**.
Description	This command sets the mobile's power control level. Notes – The RF power control level is a code. The corresponding RF output power of the mobile under test depends on the type of GSM system and the frequency range. – The power control level is broadcast in the SACCH physical header.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**CONFigure:GSM:MSTAtion:PLEVel:ALL 17** **:CONF:GSM:MSTA:PLEV?** Value returned: **"17"**.

:CONFigure:GSM:MSTAtion:MODE

Syntax	:CONFigure:GSM:MSTAtion:MODE <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **FACCh|SDCCh**. Default is **FACCh**.
Description	Defines whether the call setup takes place on **FACCh** or on **SDCCh**. **Note:** There are two standard signaling channels in GSM: SDCCH and FACCH. The FACCH is always associated with a traffic channel. However, some mobiles will only listen to signaling provided on the SDCCH. For details, please check with section Basic System Parameters.
Query	The query form of this command will return the current setting. The string delivered back will contain the short-form version of one of the predefined expressions explained above.
Example	**:CONFigure:GSM:BS:CMODe FACCh** **:CONF:GSM:BS:CMOD?** Value returned: **"FACC"**.

:CONFigure:GSM:MSTAtion:MSLot

Syntax	:CONFigure:GSM:MSTAtion:MSLot <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **1**, the maximum value is **4**. The default value for **int1** is **2**.
Description	This command either sets the number of UL time slots expected by the Willtek 3100 (asynchronous mode) or the number of UL time slots that the mobile under test is commanded to use (synchronous mode) during multislot testing. **Note:** HSCSD or multislot testing requires the Multislot Option to be installed on your Willtek 3100.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:MSTAtion:MSLot 4** **:CONF:GSM:MSTA:MSL?** Value returned: **"4"**.

:CONFigure:GSM:BER:LOOP

Syntax	:CONFigure:GSM:BER:LOOP <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **NONResidual|RESidual|FAST**. Default is **NONResidual**.
Description	This command sets the type of BER loop (inside the mobile under test) to be used for bit error measurements. **NONResidual** means that a standard bit error ratio test will be performed. This test is called **BER** on the graphical user interface of the Willtek 3100. **RESidual** will set up the test loop of a residual bit error ratio measurement. This test is called **RBER** on the graphical user interface of the Willtek 3100. **FAST** means that a fast bit error ration measurement will be performed. This test is called **FastBER** on the graphical user interface of the Willtek 3100.

Query	The query form of this command will return the current setting. The string delivered back will contain the short-form version of one of the predefined expressions explained above.
Example	**:CONFigure:GSM:BER:LOOP RESidual** **:CONF:GSM:BER:LOOP?** Value returned: **"RES"**.

:CONFigure:GSM:BER:BITPattern

Syntax	:CONFigure:GSM:BER:BITPattern <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **PRBS9|PRBS15|PRBS23|ALLZero|ALLOne| ONEZero|ZEROone**. Default is **PRBS9**.
Description	This command selects the test bit sequence to be used for the BER measurements. **PRBS9** stands for a pseudorandom bit sequence. The length of this sequence will be 511 bits ((2 to the power of 9)-1). **PRBS15** is a pseudorandom bit sequence with a length of 32767 bits. **PRBS23** is a pseudorandom bit sequence as well. The length is 8388607 bits. **ALLZero** means that all bits used for testing are 'zeros' (**000...**). In case, **ALLOne** is selected, all bits used for testing will be set to 'ones' (**111...**). **ONEZero** stands for a bit sequence starting with a 'one' and continuing with the inverted value of the current bit (**10101...**). **ZEROone** means a bit sequence starting with a 'zero' and continuing with the inverted value of the current bit (**01010...**).
Query	The query form of this command will return the current setting. The string delivered back will contain the short-form version of one of the predefined expressions explained above.
Example	**:CONFigure:GSM:BER:BITPattern ALLZero** **:CONF:GSM:BER:BITP?** Value returned: **"ALLZ"**.

:CONFigure:GSM:BER:COUNt

Syntax	:CONFigure:GSM:BER:COUNt <long1>
Parameters	**long1** is a long integer number. The minimum value for **long1** is **2,000**, the maximum value is **1,000,000**. The default value for **long1** is **10,000**.
Description	This command sets the number of bits that shall be used for a BER measurement.
Query	The query form of this command will return the current setting. The string delivered back will contain one long integer number.
Example	**:CONFigure:GSM:BER:COUNt 100000** **:CONF:GSM:BER:COUN?** Value returned: **"100000"**.

:CONFigure:GSM:BER:RTDelay

Syntax	`:CONFigure:GSM:BER:RTDelay <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **30**. The default value for **int1** is **0**.
Description	This command allows to predefine a round trip delay in order to achieve a fast synchronization. **Note:** During BER testing, there will be a delay between the reception of the first burst of a particular voice block at the mobile and the transmission of the first burst of the same voice block (after decoding and coding) on the uplink. This delay is called the round trip delay.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:CONFigure:GSM:BER:RTDelay 9** **:CONF:GSM:BER:RTD?** Value returned: **"9"**

:CONFigure[:GSM]:MEASure:GROUp[:RFTX]

Syntax	`:CONFigure[:GSM]:MEASure:GROUp[:RFTX] <PredefExp1>`
	`,[PredefExp2],[PredefExp3],[PredefExp4]`
	`,[PredefExp5],[PredefExp6],[PredefExp7]`
	`,[PredefExp8],[PredefExp9]`

Parameters	**PredefExpx** is one of the following predefined expressions: **PPEAk|PRMS|FREQuency|LENGth|UTIMe|POWer|TEMPlate|CORNer|FLATness**. Default is **PPEAK**.
Description	This command allows to set a user-specific sequence of RF TX tests on the Willtek 3100. The sequence has to be started with the appropriate commands of the **MEASure** subsystem (**:MEAS:CONT:RFTX:GROU** or **:MEAS:ARR:RFTX:GROU**). The meaning of the parameters is as follows: **PPEAk** is the peak phase error measurement, **PRMS** the root-mean square value of the phase error measurement. **FREQuency** means the frequency error of the mobile under test, **LENGth** the length of the burst in microseconds. **UTIMe** stands for the timing error of the mobile under test in microseconds, taking the ordered timing advance into account. **POWer** is the mobile's RF output power level in dBm. **TEMPlate** indicates whether the power/time template has been violated by a burst or not. **CORNER** gives the RF power level in dBm at eight predefined positions of a burst (see section Corner Points for details). **FLATness** gives an indication whether the mobile's RF output power is constant during the useful part of the burst. Notes – This command must be used with at least one parameter specified. All other parameters are optional. – Unless organized in this sequence, the start of a new RF TX test will always terminate the preceding one instantly. Therefore, this command is an efficient way to perform several RF TX measurements in a sequence.

Query	The query form of this command will return the current sequence of RF TX tests to be performed by the Willtek 3100. The string delivered back will contain a maximum of nine short-form versions of the predefined expressions explained above.
Example	**:CONFigure:GSM:MEASure:GROUp:RFTX PRMS,POWer,FLATness** **:CONF:MEAS:GROU?** Value returned: **"PRMS,POW,FLAT"**

:CONFigure[:GSM]:MEASure:GROUp:AFANalyser

Syntax	:CONFigure[:GSM]:MEASure:GROUp:AFANalyser <PredefExp1>,[PredefExp2],[PredefExp3] ,[PredefExp4],[PredefExp5],[PredefExp6]
Parameters	**PredefExpx** is one of the following predefined expressions: **SINad|DISTortion|FREQuency|ACVPeakp|** **ACVRms|DCVRms**. Default is **SINad**.
Description	This command allows to set a user-specific sequence of AF tests on the Willtek 3100. The sequence has to be started with the appropriate commands of the **MEASure** subsystem (**:MEAS:CONT:AFAN:GROU** or **:MEAS:ARR:AFAN:GROU**). The meaning of the parameters is as follows: **SINad** is the signal to noise ratio, expressed in dB, **DISTortion** is the distortion measurement on the third harmonic of a sine wave and expressed in %. **FREQuency** is the measurement of the audio frequency. **ACVPeakp** stands for an AC voltage peak-to-peak measurement, while **ACVRms** is the same measurement, but the result will be expressed as root-mean square value of the AC voltage. **DCVRms** is used to measure ripple on DC lines (this measurement gives the root-mean spare voltage of the AC component of the applied DC signal). Notes – This command must be used with at least one parameter specified. All other parameters are optional. – Unless organized in this sequence, the start of a new AF test will always terminate the preceding one instantly. Therefore, this command is an efficient way to perform several AF measurements in a sequence. – Please note that any AF test command will need the audio option to be installed on your Willtek 3100.
Query	The query form of this command will return the current sequence of AF tests to be performed by the Willtek 3100. The string delivered back will contain a maximum of six predefined short-form versions of the expressions explained above.
Example	**:CONFigure:GSM:MEASure:GROUp:AFANalyser SINad,DISTortion,FREQuency** **:CONF:MEAS:GROU:AFAN?** Value returned: **"SIN,DIST,FREQ"**

:CONFigure:MEASure:GROUp:PSUPply

Syntax	:CONFigure:MEASure:GROUp:PSUPply <PredefExp1>,[PredefExp2],[PredefExp3]
Parameters	**PredefExpx** is one of the following predefined expressions: **CAVG|CPEak|PAVG**. Default is **CAVG**.
Description	This command allows to set a user-specific sequence of power supply tests on the Willtek 3100. The sequence has to be started with the appropriate commands of the **MEASure** subsystem (**:MEAS:CONT:PSUP:GROU** or **:MEAS:ARR:PSUP:GROU**). The meaning of the parameters is as follows: **CAVG** stands for the average current drawn by the mobile, while **CPEak** is the peak current drawn by the mobile over the defined measurement period. The defined measurement period can be set using the command **:PSUP:MEAS:DURA**. **PAVG** is the average power requirement of the mobile under test. Notes – This command must be used with at least one parameter specified. All other parameters are optional. – Unless organized in this sequence, the start of a new power supply test will always terminate the preceding one instantly. Therefore, this command is an efficient way to perform several power supply measurements in a sequence. – Please note that any power supply test command will need the power supply option to be installed on your Willtek 3100.
Query	The query form of this command will return the current sequence of power supply tests to be performed by the Willtek 3100. The string delivered back will contain a maximum of three short-form versions of the predefined expressions explained above.
Example	**:CONFigure:MEASure:GROUp:PSUPply CPEak,PAVG** **:CONF:MEAS:GROU:PSUP?** Value returned: **"CPE,PAVG"**

:CONFigure:GSM:MEASure:ACPM:TRANsient

Syntax	:CONFigure:GSM:MEASure:ACPM:TRANsient <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **EDGes|FULL**. Default is **EDGes**.
Description	Selects the time period over which the Willtek 3100 calculates the RF output spectrum due to switching transients (or ACP Transient for short). If **FULL** is selected, the Willtek 3100 will perform the measurement over the whole burst (including the edges); this is in line with the ETSI specifications. With **EDGes**, measurements are performed over the leading and trailing edges of the burst only, ignoring any spectral impact of the modulation (active part of the burst). The latter measurement takes less time and hence leads to faster measurements.
Query	The query form of this command will return the current setting. The string delivered back will contain the short-form version of one of the predefined expressions explained above.

Example	**:CONF:GSM:MEASure:ACPM:TRAN FULL** **:CONF:GSM:MEASure:ACPM:TRAN?** In this example, the query will return the string **FULL**.

:CONFigure:GSM:MEASure:LEVel:EXPect

Syntax	:CONFigure:GSM:MEASure:LEVel:EXPect <real1>
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **-16.0**, the maximum value **30.0**. The minimum resolution possible for **real1** is **0.1**. The default value for **real1** is **0.0**.
Description	This level is expected for Fast Power Measurement (avoids delays due to gain control)
Query	The query form of this command will return the currently expected level for Fast Power Measurement
Example	:CONFigure[:GSM]:MEASure:LEVel:EXPect 7 :CONFigure[:GSM]:MEASure:LEVel:EXPect? Value returned: "7".

:CONFigure:COUPloss:STATe

Syntax	:CONFigure:COUPloss:STATe <PredefExp1>
Parameters	**PredefExp1** is one of the following predefined expressions: **ON|OFF**. Default is **OFF**.
Description	This command switches the use of a coupling loss table either on or off. **Note:** Coupling loss tables are used to compensate e.g. losses in cables. For more details, please refer to section Coupling Loss.
Query	The query form of this command will return the current setting. The string delivered back will contain one predefined expression as explained above.
Example	**:CONFigure:COUPloss:STATe ON** **:CONF:COUP:STAT?** Value returned: **"ON"**

:CONFigure:COUPloss:NAME

Syntax	:CONFigure:COUPloss:NAME <string1>
Parameters	**string1** is as string, giving the complete file name (and directory) of the coupling loss file to be loaded. The maximum length of **string1** is 50 characters. The default for **string1** is **"example.cpl"**.
Description	This command loads the coupling loss description file, specified with the command's parameter. Please note that the data contained in the file need to be activated (using the **CONF:COUP:STAT ON** command described above) before the data contained in the file specified will have any effect on the measurement results. For more details, please refer to section "Coupling Loss".
Query	The query form of this command will return the name of the currently loaded coupling loss description file. The string delivered back will contain the full file name.

Example	**:CONFigure:COUPloss:NAME "m7389.cpl"** **:CONF:COUP:NAME?** Value returned: **"m7389.cpl"**

:CONFigure:COUPloss:INFormation?

Syntax	:CONFigure:COUPloss:INFormation?
Parameters	There are no parameters.
Description	This command is used to read out the comments saved with the coupling loss description file currently loaded. **Note:** There is only a query form of this command available.
Query	The query form of this command will return the comments saved with the coupling loss description. The string delivered back will contain a maximum of 255 characters.
Example	**:CONFigure:COUPloss:INFormation?** Value returned: **"Motorola P7389 with Antenna Coupler"**

:CONFigure:COUPloss:DATA

Syntax	:CONFigure:COUPloss:DATA <string1> ,<realf1>,<reala1>,<realf2>,<reala2> [,realf3] [,reala3] [,realf4] [,reala4] [,realf5] [,reala5] [,realf6] [,reala6] [,realf7] [,reala7] [,realf8] [,reala8] [,realf9] [,reala9] [,realf10] [,reala10] [,realf11] [,reala11] [,realf12] [,reala12] [,realf13] [,reala13] [,realf14] [,reala14] [,realf15] [,reala15] [,realf16] [,reala16] [,realf17] [,reala17] [,realf18] [,reala18] [,realf19] [,reala19] [,realf20] [,reala20]
Parameters	**string1** is a comment line related to the coupling loss data. The maximum length of **string1** is 255 characters. **realfx** are floating point real numbers, giving frequencies in MHz while **realax** are floating point real numbers, giving the corresponding coupling loss in dB. All **realfx** have to be within two frequency ranges. The lower frequency range is from **800.0** MHz to **1000.0** MHz; the higher frequency range is from **1700.0** MHz to **2000.0** MHz. The minimum resolution for all **realfx** values is 10 Hz (**0.00001** MHz). The default value for all **realfx** is **800.0** MHz. The minimum value for all **realax** is **-5.0** dB. The maximum value for all **realax** is **40.0** dB. The minimum resolution for all **realax** is **0.01** dB. The default value for all **realax** is **0.0**. **Notes** – Please keep in mind that at least one pair of values for a frequency and the related attenuation **must** be specified **per frequency range**, while all other 18 pairs are optional. – All **realax** values are interpreted as an attenuation level in dB. As a consequence, negative values mean an amplification of the input signal.
Description	With the help of this command, you may create a coupling loss description table, similar to how you would do it on the graphical user interface of the Willtek 3100.
Query	There is no query form of this command available.

Example	`:CONFigure:COUPloss:DATA "Motorola 7389 with cable #23",825.0,15.0,1750.0,19.0`

:CONFigure:ESYNc

Syntax	`:CONFigure:ESYNc?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form will return the status of the current external synchronization. The string delivered back will contain one of the following expressions: `NONE|MHZ5|MHZ10|MHZ13`.
Example	`:CONFigure:ESYNc?` Value returned for example: `"MHZ10"`. In this example, the 3100 frequency reference is synchronized to an external 10 MHz clock signal fed into the EXT SYNC connector on the rear panel.

:CONFigure:WCDMa:POWMeasure:RRCFilter

Syntax	`:CONFigure:WCDMa:POWMeasure:RRCFilter <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON|OFF`. The default is `ON`.
Description	This command sets the state of the RRC Filter for power measurements.
Query	The query form of this command returns the current setting for `PredefExp`.

Example	`:CONF:WCDMA:POWM:RRCF OFF`
	`:CONF:WCDMA:POWM:RRCF?`
	`Value returned in this example: OFF.`

:CONFigure:WCDMa:NB:LEVel

Syntax	`:CONFigure:WCDMa:NB:LEVel <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-120`, the maximum value `-20`. The minimum resolution is `0.1`. The default value is `-40`.
Description	This command sets the RF output level of the 3100. The value specified for `real1` is the power output level in dBm.
Query	The query form of this command will return the current setting of the Willtek 3100.

Example	`:CONF:WCDMA:NB:LEV -38.0`
	`:CONF:WCDMA:NB:LEV?`
	Value returned in this example: `-38.0`.

:CONFigure:WCDMa:NB:LEVel:RELative

Syntax	`:CONFigure:WCDMa:NB:LEVel:RELative <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `CPICh|IOR`. The default is `CHICh`.
Description	This command sets the power type which each code channel is relative to.
Query	The query form of this command returns the current setting for `PredefExp`.

Example	`:CONF:WCDMA:NB:LEV:REL IOR`
	`:CONF:WCDMA:NB:LEV:REL?`
	Value returned in this example: `IOR`.

:CONFigure:WCDMa:NB:LEVel:CPICh

Syntax	`:CONFigure:WCDMa:NB:LEVel:CPICh <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-35`, the maximum value is `0`. The minimum resolution is `0.1`. The default value is `-4.2`.
Description	This command sets the Primary CPICH power relative to Ior. This is an exception, because this special code channel is always defined relative to Ior, regardless of what is set with `CONF:WCDM:LEV:REL`.
Query	The query form of this command will return the current setting of the Willtek 3100.

Example	`:CONF:WCDMA:NB:LEV:CPIC -30`
	`:CONF:WCDMA:NB:LEV:CPIC?`
	Value returned in this example: `-30`.

:CONFigure:WCDMa:NB:LEVel:DPCH

Syntax	`:CONFigure:WCDMa:NB:LEVel:DPCH <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-35`, the maximum value is `0`. The minimum resolution is `0.1`. The default value is `-5.0`.
Description	This command sets the DPCH power relative to CPICH or Ior, depending on what is set with `CONF:WCDM:LEV:REL`.
Query	The query form of this command will return the current setting of the Willtek 3100.

Example	`:CONF:WCDMA:NB:LEV:DPCH -20`
	`:CONF:WCDMA:NB:LEV:DPCH?`
	Value returned in this example: `-20`.

:CONFigure:WCDMa:NB:LEVel:PCCPch

Syntax	`:CONFigure:WCDMa:NB:LEVel:PCCPch <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-35`, the maximum value is `0`. The minimum resolution is `0.1`. The default value is `-6.0`.
Description	This command sets the PCCPCH power relative to CPICH or Ior, depending on what is set with `CONF:WCDM:LEV:REL`.
Query	The query form of this command will return the current setting of the Willtek 3100.

Example	`:CONF:WCDMA:NB:LEV:PCCP -10`
	`:CONF:WCDMA:NB:LEV:PCCP?`
	Value returned in this example: `-10`.

:CONFigure:WCDMa:NB:LEVel:SCCPch

Syntax	`:CONFigure:WCDMa:NB:LEVel:SCCPch <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-35`, the maximum value is `0`. The minimum resolution is `0.1`. The default value is `-3.0`.
Description	This command sets the SCCPCH power relative to CPICH or Ior, dependent from what is set with `CONF:WCDM:LEV:REL`.
Query	The query form of this command will return the current setting of the Willtek 3100.

Example	`:CONF:WCDMA:NB:LEV:SCCP -15`
	`:CONF:WCDMA:NB:LEV:SCCP?`
	Value returned in this example: `-15`.

:CONFigure:WCDMa:NB:LEVel:PSCH

Syntax	`:CONFigure:WCDMa:NB:LEVel:PSCH <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-35`, the maximum value is `0`. The minimum resolution is `0.1`. The default value is `-3.0`.
Description	This command sets the PSCH power relative to CPICH or Ior, depending on what is set with `CONF:WCDM:LEV:REL`.
Query	The query form of this command will return the current setting of the Willtek 3100.

Example	`:CONF:WCDMA:NB:LEV:PSCH -12`
	`:CONF:WCDMA:NB:LEV:PSCH?`
	Value returned in this example: `-12`.

:CONFigure:WCDMa:NB:LEVel:PICH

Syntax	`:CONFigure:WCDMa:NB:LEVel:PICH <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-35`, the maximum value is `0`. The minimum resolution is `0.1`. The default value is `0.0`.
Description	This command sets the PICH power relative to CPICH or Ior, depending on what is set with `CONF:WCDM:LEV:REL`. Also this parameter range depends on the setting from `CONF:WCDM:LEV:REL`. Its minimum is `-35.0` in case that it is relative to Ior. If it is relative to CPICH, however, then it is `-10.0`.
Query	The query form of this command will return the current setting of the Willtek 3100.

Example	`:CONF:WCDMA:NB:LEV:PICH -16`
	`:CONF:WCDMA:NB:LEV:PICH?`
	Value returned in this example: `-16`.

:CONFigure:WCDMa:NB:LEVel:AICH

Syntax	`:CONFigure:WCDMa:NB:LEVel:AICH <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-35`, the maximum value is `0`. The minimum resolution is `0.1`. The default value is `-0.6`.
Description	This command sets the AICH power relative to CPICH or Ior, depending on what is set with `CONF:WCDM:LEV:REL`.
Query	The query form of this command will return the current setting of the Willtek 3100.

Example	`:CONF:WCDMA:NB:LEV:AICH -17`
	`:CONF:WCDMA:NB:LEV:AICH?`
	Value returned in this example: `-17`.

:CONFigure:WCDMa:NB:LEVel:OCNS

Syntax	`:CONFigure:WCDMa:NB:LEVel:OCNS <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-35`, the maximum value is `0`. The minimum resolution is `0.1`. The default value is `-5.0`.
Description	This command sets the OCNS power relative to CPICH or Ior, depending on what is set with `CONF:WCDM:LEV:REL`.
Query	The query form of this command will return the current setting of the Willtek 3100.

Example	`:CONF:WCDMA:NB:LEV:OCNS -18`
	`:CONF:WCDMA:NB:LEV:OCNS?`
	Value returned in this example: `-18`.

:CONFigure:WCDMa:NB:LEVel:OCNS:STATe

Syntax	:CONFigure:WCDMa:NB:LEVel:OCNS:STATe <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: ON|OFF. The default is OFF.
Description	This command sets the OCNS state on or off.
Query	The query form of this command returns the current setting for PredefExp.
Example	:CONF:WCDMA:NB:LEV:OCNS:STAT OFF :CONF:WCDMA:NB:LEV:OCNS:STAT? Value returned in this example: OFF.

:CONFigure:WCDMA:NB:CELL:MCC

Syntax	:CONFigure:WCDMA:NB:CELL:MCC <Int1>
Parameters	Int1 is an integer number. The minimum value is 0, the maximum value 999. The default value is 1.
Description	This command sets the mobile country code (MCC).
Query	The query form of this command will return the current setting of the Willtek 3100. The string delivered will contain one integer number.
Example	:CONF:WCDMA:NB:CELL:MCC 1 :CONF:WCDMA:NB:CELL:MCC? Value returned in this example: 1.

:CONFigure:WCDMA:NB:CELL:MNC

Syntax	:CONFigure:WCDMA:NB:CELL:MNC <Int1>
Parameters	Int1 is an integer number. The minimum value is 0, the maximum value 999. The default value is 1.
Description	This command sets the mobile network code (MNC).
Query	The query form of this command will return the current setting of the Willtek 3100. The string delivered will contain one integer number.
Example	:CONF:WCDMA:NB:CELL:MNC 0 :CONF:WCDMA:NB:CELL:MNC? Value returned in this example: 0.

:CONFigure:WCDMA:NB:CELL:LAC

Syntax	:CONFigure:WCDMA:NB:CELL:LAC <Int1>
Parameters	Int1 is an integer number. The minimum value is 0, the maximum value 65535. The default value is 4660.
Description	This command sets the UE's LAC.
Query	The query form of this command will return the current setting of the Willtek 3100. The string delivered will contain one integer number.
Example	:CONF:WCDMA:NB:CELL:LAC 0 :CONF:WCDMA:NB:CELL:LAC? Value returned in this example: 0.

:CONFigure:WCDMA:NB:CELL:URA

Syntax	:CONFigure:WCDMA:NB:CELL:URA <Int1>
Parameters	Int1 is an integer number. The minimum value is 0, the maximum value 65535. The default value is 29136.
Description	This command sets the UE's URA.
Query	The query form of this command will return the current setting of the Willtek 3100. The string delivered will contain one integer number.
Example	:CONF:WCDMA:NB:CELL:URA 30000 :CONF:WCDMA:NB:CELL:URA? Value returned in this example: 30000.

:CONFigure:WCDMA:NB:CELL:CIDentity

Syntax	:CONFigure:WCDMA:NB:CELL:CIDentity <Int1>
Parameters	Int1 is an integer number. The minimum value is 0, the maximum value 216494545. The default value is 216494544.
Description	This command sets the UE's cell identity.
Query	The query form of this command will return the current setting of the Willtek 3100. The string delivered will contain one integer number.
Example	:CONF:WCDMA:NB:CELL:CID 216494545 :CONF:WCDMA:NB:CELL:CID? Value returned in this example: 216494545.

:CONFigure:WCDMA:NB:CELL:PSC

Syntax	`:CONFigure:WCDMA:NB:CELL:PSC <Int1>`
Parameters	`Int1` is an integer number. The minimum value is `0`, the maximum value `511`. The default value is `0`.
Description	This command sets the NB primary scrambling code.
Query	The query form of this command will return the current setting of the Willtek 3100. The string delivered will contain one integer number.

Example	`:CONF:WCDMA:NB:CELL:CID 216494545`
	`:CONF:WCDMA:NB:CELL:CID?`
	Value returned in this example: `216494545`.

:CONFigure:WCDMa:NB:CELL:IMSI2

Syntax	`:CONFigure:WCDMa:NB:CELL:IMSI2 <string1>`
Parameters	`string1` is a string indicating the IMSI2 number. The maximum length of `string1` is 19 characters. The default for `string1` is empty.
Description	This command sets the UE's secondary IMSI which is customer specific.
Query	The query form of this command will return the current setting.

Example	`:CONF:WCDMA:NB:CELL:IMSI2 "255255255255255255"`
	`:CONF:COUP:NAME?`
	Value returned in this example: `"255255255255255255"`

:CONFigure:WCDMa:NB:RMC

Syntax	`:CONFigure:WCDMa:NB:RMC <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `RMC12|RMC64|RMC144|RMC384`. The default is `RMC12`.
Description	This command sets the Reference Measurement Channel.
Query	The query form of this command returns the current setting for `PredefExp`.

Example	`:CONF:WCDMA:NB:RMC RMC384`
	`:CONF:WCDMA:NB:RMC?`
	Value returned in this example: `RMC384`.

:CONFigure:WCDMa:NB:ATTach

Syntax	`:CONFigure:WCDMa:NB:ATTach <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `OFF|ON`. The default is `OFF`.
Description	Forces the UE to re-register even though it was registered to the cell before getting switched off.

Query	The query form of this command returns the current setting for `PredefExp`.

Example	`:CONF:WCDMA:NB:ATT ON`
	`:CONF:WCDMA:NB:ATT?`
	Value returned in this example: `ON`.

:CONFigure:WCDMa:NB:SPLoop

Syntax	`:CONFigure:WCDMa:NB:SPLoop <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `OFF|ON`. The default is `OFF`.
Description	Activates or disactivates the speech loopback mechanism to give an audible indication of the speech call.
Query	The query form of this command returns the current setting for `PredefExp`.

Example	`:CONF:WCDMA:NB:SPL ON`
	`:CONF:WCDMA:NB:SPL?`
	Value returned in this example: `ON`.

:CONFigure:WCDMa:NB:BAND

Syntax	`:CONFigure:WCDMa:NB:BAND <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `BAND1|BAND2|BAND3|BAND4|BAND5|BAND6|BAND14|BAND34`. The default is `BAND1`.
Description	This command sets the actual WCDMA operating band.
Query	The query form of this command returns the current setting for `PredefExp`.
Example	`:CONF:WCDMA:NB:BAND BAND2` Value returned in this example: `BAND2`.

:CONFigure:WCDMa:NB:DLUarfcn

Syntax	`:CONFigure:WCDMa:NB:DLUarfcn <Int1>`
Parameters	`Int1` is an integer number. The minimum value is `412`, the maximum value `10838`. The default value is `10700`.
Description	This command sets the WCDMA Downlink channel. **Note:** This command also sets the Uplink channel to the corresponding channel.
Query	The query form of this command will return the current setting of the Willtek 3100. The string delivered back will contain one integer number.

Example	`:CONF:WCDMA:NB:DLU 5000`
	`:CONF:WCDMA:NB:DLU?`
	Value returned in this example: `5000`.

:CONFigure:WCDMa:NB:ULUarfcn

Syntax	:CONFigure:WCDMa:NB:ULUarfcn <Int1>
Parameters	Int1 is an integer number. The minimum value is 12, the maximum value 9888. The default value is 9750.
Description	This command sets the WCDMA Uplink channel. **Note:** This command also sets the Downlink channel to the corresponding channel.
Query	The query form of this command will return the current setting of the Willtek 3100. The string delivered will contain one integer number.
Example	:CONF:WCDMA:NB:ULU 5000 :CONF:WCDMA:NB:ULU? Value returned in this example: 5000.

:CONFigure:WCDMa:MEASure:UEReport:TRCH

Syntax	:CONFigure:WCDMa:MEASure:UEReport:TRCH
Parameters	PredefExp is one of the following predefined expressions: RAB1|RAB2|RAB3|SRB. The default is RAB1.
Description	Defines on which kind of Transport Channel the BLER estimation within the UE applies.
Query	The query form of this command returns the measurement that has been choosen.
Example	:CONF:WCDMA:MEAS:UEReport:TRCH RAB2 :CONF:WCDMA:MEAS:UEReport:TRCH? Returned string in this example: RAB2.

:CONFigure:WCDMa:MEASure:RFRX:NBLOck

Syntax	:CONFigure:WCDMa:MEASure:RFRX:NBLOck <Int1>
Parameters	Int1 is an integer number. The minimum value is 10, the maximum value 4096. The default value is 40.
Description	This command sets the number of blocks to use in the BER/BLER test.
Query	The query form of this command will return the current setting of the Willtek 3100.
Example	:CONF:WCDM:MEAS:RFRX:NBLO 1000 :CONF:WCDMA:MEAS:RFRX:NBLO? Value returned in this example: 1000.

:CONFigure:WCDMa:MEASure:RFRX:BITPattern

Syntax	:CONFigure:WCDMa:MEASure:RFRX:BITPattern <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: PRBS9|PRBS15|PRBS23|ALLZero|ALLOne|ONEZero|ZEROone. Default is PRBS9.
Description	This command sets the bit pattern for the WCDMA signal.

Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.

Example	`:CONF:WCDM:MEAS:RFRX:BITP ONEZero`
	`:CONF:WCDM:MEAS:RFRX:BITP?`
	Value returned in this example "`ONEZero`".

:CONFigure:COUPloss:STATe

Syntax	`:CONFigure:COUPloss:STATe <PredefExp1>`
Parameters	`PredefExp1` is one of the following predefined expressions: `ON|OFF`. Default is `OFF`.
Description	This command switches the use of a coupling loss table either `ON` or `OFF`. **Note:** Coupling loss tables are used to compensate e.g. losses in cables. For more details, please refer to section Coupling Loss.
Query	The query form of this command will return the current setting. The string delivered will contain one predefined expression as explained above.

Example	`:CONFigure:COUPloss:STATe ON`
	`:CONF:COUP:STAT?`
	Value returned in this example: "`ON`"

:CONFigure:COUPloss:NAME

Syntax	`:CONFigure:COUPloss:NAME <string1>`
Parameters	`string1` is a string, specifying the complete file name (and directory) of the coupling loss file to be loaded. The maximum length of `string1` is 50 characters. The default for `string1` is "`example.cpl`".
Description	This command loads the coupling loss description file, specified with the command's parameter. Please note that the data contained in the file need to be activated (using the `CONF:COUP:STAT ON` command described above) before the data contained in the file specified will have any effect on the measurement results. For more details, please refer to section Coupling Loss.
Query	The query form of this command will return the name of the currently loaded coupling loss description file. The string delivered will contain the full file name.

Example	`:CONFigure:COUPloss:NAME "m7389.cpl"`
	`:CONF:COUP:NAME?`
	Value returned in this example: "`m7389.cpl`"

:CONFigure:COUPloss:INFormation?

Syntax	`:CONFigure:COUPloss:INFormation?`
Parameters	There are no parameters.
Description	This command is used to read out the comments saved with the coupling loss description file currently loaded. For more details, please refer to section Coupling Loss. Note There is only a query form of this command available.

Query	The query form of this command will return the comments saved with the coupling loss description. The string delivered back will contain a maximum of 255 characters.
Example	`:CONFigure:COUPloss:INFormation?` Value returned in this example: `"Motorola P7389 with Universal Antenna Coupler"`

:CONFigure:WCDMa:COUPloss:DATA

Syntax	`:CONFigure:WCDMa:COUPloss:DATA <string1>`
	`,<realf1>,<reala1>,<realf2>,<reala2> [,realf3]`
	`[,reala3] [,realf4] [,reala4] [,realf5] [,reala5]`
	`[,realf6] [,reala6] [,realf7] [,reala7] [,realf8]`
	`[,reala8] [,realf9] [,reala9] [,realf10] [,reala10]`
	`[,realf11] [,reala11] [,realf12] [,reala12] [,realf13]`
	`[,reala13] [,realf14] [,reala14] [,realf15] [,reala15]`
	`[,realf16] [,reala16] [,realf17] [,reala17] [,realf18]`
	`[,reala18] [,realf19] [,reala19] [,realf20] [,reala20]`

Parameters	`string1` is a comment line related to the coupling loss data. The maximum length of `string1` is 255 characters. `realfx` are floating point real numbers, giving frequencies in MHz while `realax` are floating point real numbers, giving the corresponding coupling loss in dB. All `realfx` have to be within two frequency ranges. The lower frequency range is from `800.0` MHz to `1000.0` MHz, the higher frequency range is from `1700.0` MHz to `2000.0` MHz. The minimum resolution for all realfx values is `10` Hz (`0.00001` MHz). The default value for all `realfx` is `800.0` MHz. The minimum value for all `realax` is `-5.0` dB. The maximum value for all `realax` is `40.0` dB. The minimum resolution for all `realax` is `0.01` dB. The default value for all `realax` is `0.0`. **Notes** – Please keep in mind that at least one pair of values for a frequency and the related attenuation **must** be specified **per frequency range**, while all other 18 pairs are optional. – All `realax` values are interpreted as an attenuation level in dB. As a consequence, negative values mean an amplification of the input signal.
Description	With the help of this command, you may create a coupling loss description table, similar to how you would do it on the graphical user interface of the Willtek 3100. For further details, please refer to section Coupling Loss.
Query	There is no query form of this command available.
Example	`:CONF:WCDMA:COUP:DATA "Motorola 7389 with cable #23",825.0,15.0,1750.0,19.0`

CALL subsystem

This subsystem contains commands

- for call setup and handling procedures and
- to read out the measurement report, generated by the mobile.

:CALL:GSM:BSORiginate

Syntax	:CALL:GSM:BSORiginate
Parameters	There are no parameters.
Description	Initiates a call-setup by the base station (Willtek 3100). For details regarding a BS initiated call, please refer to section BS Call.
Query	There is no query form of this command available.
Example	**:CALL:GSM:BSOR** This command will initiate a BS-originated call.

:CALL:GSM:BSORiginate:SILEnt

Syntax	:CALL:GSM:BSORiginate:SILEnt
Parameters	There are no parameters.
Description	Initiates a call-setup by the base station (Willtek 3100) without alerting.
Query	There is no query form of this command available.
Example	:CALL:GSM:BSOR:SILE This command will initiate a BS-originated call without alerting.

:CALL:GSM:RSIGnaling

Syntax	:CALL:GSM:RSIGnaling
Parameters	There are no parameters.
Description	Makes the Willtek 3100 use reduced signaling. This means that the test set will provide a base channel and react to the mobile's signals without prior call setup.
Query	There is no query form of this command available.
Example	:CALL:GSM:RSIG

:CALL:GSM:BSRelease

Syntax	:CALL:GSM:BSRelease
Parameters	There are no parameters.
Description	This command will make the Willtek 3100 terminate the ongoing call. For details regarding a BS-terminated call, please refer to section BS Call.

Query	There is no query form of this command available.
Example	**:CALL:GSM:BSR** This command will release the call.

:CALL:GSM:MSORiginate

Syntax	:CALL:GSM:MSORiginate
Parameters	There are no parameters.
Description	This command initiates the MS call procedure on the Willtek 3100. For details regarding the MS call procedure, please refer to section MS Call. **Note:** This command is no longer required. This means that the Willtek 3100 will automatically respond to the mobile's channel request.
Query	There is no query form of this command available.
Example	As this command is no longer required, an example was skipped.

:CALL:GSM:MSRelease

Syntax	:CALL:GSM:MSRelease
Parameters	There are no parameters.
Description	This command prepares the call termination by the mobile under test. For details regarding the MS call procedure, please refer to section MS Call. **Note:** This command is no longer required.
Query	There is no query form of this command available.
Example	As this command is no longer required, an example was skipped.

:CALL:GSM:PAGing

Syntax	:CALL:GSM:PAGing <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **OFF**.
Description	Switches the continuous paging of the mobile **ON** or **OFF**. **ON** means that the Willtek 3100 will start to broadcast paging requests, but will not answer to the mobile's access bursts. For details regarding continuous paging, please refer to section Call Graph.
Query	There is no query form of this command available.
Example	**:CALL:GSM:PAG ON** This command switches continuous paging on.

:CALL:GSM:ASSignment

Syntax	:CALL:GSM:ASSignment <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **LATe|EARLy**. Default is **LATe**.
Description	Determines whether early or late assignment shall be used. For details regarding assignment, please refer to section Basic System Parameters.
Query	There is no query form of this command available.
Example	**:CALL:GSM:ASS EARL** Early assignment will be used.

:CALL:GSM:SMS:MSOR

Syntax	:CALL[:GSM]:SMS:MSOR <string1>
Parameters	There are no parameters.
Description	This command reads out the text and the phone number of the last received short message.
Query	There is only a query form of this command available. The query will return a string containing 2 values coded as explained above.
Example	:CALL:GSM:SMS:MSOR? The string delivered back in this example is: ""3100 Mobile Fault Finder","+4989996410"".

:CALL[:GSM]:SMS:MSOR:TEXT

Syntax	:CALL[:GSM]:SMS:MSOR:TEXT <string1>
Parameters	There are no parameters.
Description	This command reads out the text of the last received short message.
Query	There is only a query form of this command available. The query will return a string containing a value coded as explained above.
Example	:CALL:GSM:SMS:MSOR:TEXT? The string delivered in this example is: ""3100 Mobile Fault Finder"".

:CALL[:GSM]:SMS:MSOR:NUMB

Syntax	:CALL[:GSM]:SMS:MSOR:NUMB <string1>
Parameters	There are no parameters.
Description	This command reads out the phone number of the last received short message.
Query	There is only a query form of this command available. The query will return a string containing a value coded as explained above.
Example	:CALL:GSM:SMS:MSOR:NUMB? The string delivered back in this example is: ""+4989996410"".

:CALL[:GSM]:SMS:BSOR

Syntax	:CALL[:GSM]:SMS:BSOR <string1>
Parameters	There are no parameters.
Description	This command sends a short message.
Query	A query form of this command is not available.
Example	:CALL:GSM:SMS:BSOR "Willtek Mobile Tester","12345" This command will send a BS-originated short message.

:CALL:GSM:BERLoop

Syntax	:CALL:GSM:BERLoop <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **OFF**.
Description	This command is used to close or open the internal loop of the mobile for BER testing. **ON** means that the Willtek 3100 will tell the mobile under test to close its internal loop, while **OFF** will make the Willtek 3100 tell the mobile under test to open the internal loop. For details regarding BER testing, please refer to section BER Measurements.
Query	The query form of this command will return the current setting. The string delivered back will contain one of the predefined expressions explained above.
Example	**:CALL:GSM:BERL ON** **:CALL:BERL?** Value returned: "ON".

:CALL:GSM:MSINfo:ALL

Syntax :CALL:GSM:MSINfo:ALL

Parameters There are no parameters.

Description This command reads out all MS Info result values with one query. For details on MS Info, please refer to section MS Info Test Menu.
The query will deliver back 15 single values in the following sequence, all separated by commas:

- **RX Lev (Integer)** The mobile estimates the RF power level at which it receives the base station. The coding of this figure can be found in section Coding of RX Lev.
- **RX Qual (Integer)** Based on the number of bits corrected by its own channel decoder, the mobile estimates the BER. The coding of this figure can be found in section Coding of RX Qual.
- **Number (String)** This position contains the number dialed on the mobile. In case the call was BS originated, this position will be left empty (i.e. two consecutive commas in the result string).
- **IMSI (String)** This is the international mobile subscriber identity.
- **IMEI (String)** This position contains the international mobile equipment identity.
- **MS Class (Integer)** Shows the mobile's power class. The coding of this figure can be found in section MS Power Classes.
- **Rev. Level (Integer)** This parameter indicates whether the mobile supports all features of GSM phase 1 or phase 2 or any later phase.
- **RX Lev Sub (Integer)** Similar to RX Lev, but only carried out on a subset of frames. For details, please refer to section MS Report Display. The coding of the RX Lev Sub result value can be found in section Coding of RX Lev.
- **SMS (Boolean)** A **0** on this position means that the mobile under test has no SMS capabilities, while a **1** indicates that the mobile can handle SMS.
- **Ext. Freq (Boolean)** A **0** on this position means that the mobile under test can't use the extended GSM frequency range (E-GSM), while a **1** indicates that the mobile is able to work in the extended GSM frequency range.
- **A5 Cyph. (Integer)** This position shows the A5 ciphering key supported by the mobile.
- **Timing Advance (Integer)** Here, the timing advance of the mobile is shown in bit periods. More details regarding the timing advance can be found in section Timeslots and Bursts.
- **Dualband (Integer)** A **0** on this position indicates that the mobile under test is no dual-band mobile, while a **1** indicates that the mobile can work both in GSM and PCN frequency ranges.
- **MS Pwr (Integer)** Here, the power control level currently used by the mobile is shown. This figure is coded according to the scheme explained in section Power Control Levels and Tolerances.
- **EFR (Boolean)** A **0** on this position means that the mobile under test cannot use enhanced full rate TCHs, while a **1** indicates that the mobile is capable of doing so.

Query There is only a query form of this command available. The query will return a string containing 15 values coded as explained above.

Example **:CALL:GSM:MSINfo:ALL?**
The string delivered back is, for example:
"49,0,"+4989996410","001010123456789",
"520040190000430",4,2,49,1,0,3,0,1,16,1".

:CALL:GSM:MSINfo:RXLevel

Syntax	:CALL:GSM:MSINfo:RXLevel
Parameters	There are no parameters.
Description	The mobile estimates the RF power level at which it receives the base station. The coding of the result value delivered back can be found in section Coding of RX Lev.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	**:CALL:GSM:MSINfo:RXL?** The string delivered back: **"49"**.

:CALL:GSM:MSINfo:RXQual

Syntax	:CALL:GSM:MSINfo:RXQual
Parameters	There are no parameters.
Description	Based on the number of bits corrected by its own channel decoder, the mobile estimates the BER. The coding of the result value delivered back can be found in section Coding of RX Qual.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	**:CALL:GSM:MSINfo:RXQ?** The string delivered back: **"0"**.

:CALL:GSM:MSINfo:NUMBer

Syntax	:CALL:GSM:MSINfo:NUMBer
Parameters	There are no parameters.
Description	This query will deliver back the number dialed on the mobile as a string. In case the call was BS originated, the string will be empty (**""**).
Query	There is only a query form of this command available. The query will deliver back a string.
Example	**:CALL:MSINfo:NUMB?** The string delivered back: **"+4989996410"**.

:CALL:GSM:MSINfo:IMSI

Syntax	:CALL:GSM:MSINfo:IMSI
Parameters	There are no parameters.
Description	This query will deliver the international mobile subscriber identity as stored on the mobile's SIM card.
Query	There is only a query form of this command available. The query will deliver back a string.

Example	**:CALL:GSM:MSINfo:IMSI?** The string delivered back: **"001010123456789"**.

:CALL:GSM:MSINfo:IMSI:CATCH

Syntax	:CALL:GSM:MSINfo:IMSI:CATCH
Parameters	There are no parameters.
Description	This command will switch the searching for a specific IMSI either ON or OFF.
Query	There is no query form of this command available.
Example	:CALL:GSM:MSINfo:IMSI:CATCh ON

:CALL:GSM:MSINfo:IMEI

Syntax	:CALL:GSM:MSINfo:IMEI
Parameters	There are no parameters.
Description	This query will deliver back the international mobile equipment identity as stored inside the mobile (the so-called serial number).
Query	There is only a query form of this command available. The query will deliver back a string.
Example	**:CALL:GSM:MSINfo:IMSI?** The string delivered back: **"520040190000430"**.

:CALL:GSM:MSINfo:IMEI:SV

Syntax	:CALL:GSM:MSINfo:IMEI:SV
Parameters	There are no parameters.
Description	This query will deliver back the international mobile equipment identity and the software version number as stored inside the mobile.
Query	There is only a query form of this command available. The query will deliver back a string.
Example	**:CALL:GSM:MSINfo:IMSI:SV?** The string delivered back: "52004019000043-0(01)". The first part is the IMEI with the calculated check digit after the hyphen; the string in brackets indicates the software version reported over the air interface.

:CALL:GSM:MSINfo:MSCLass

Syntax	:CALL:GSM:MSINfo:MSCLass
Parameters	There are no parameters.
Description	This query will deliver back the code representing the mobile's power class. The coding of this figure can be found in section MS Power Classes.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	**:CALL:GSM:MSINfo:MSCLass?** The integer delivered back: **"4"**.

:CALL:GSM:MSINfo:RLEVel

Syntax	:CALL:GSM:MSINfo:RLEVel
Parameters	There are no parameters.
Description	This query will deliver back the mobile's GSM revision level (GSM phase **1** or phase **2** or any later phase).
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	**:CALL:GSM:MSINfo:RLEVel?** The integer delivered back: **"2"**.

:CALL:GSM:MSINfo:RXLSub

Syntax	:CALL:GSM:MSINfo:RXLSub
Parameters	There are no parameters.
Description	This query is similar to **:CALL:MSINfo:RXLevel?** as explained above. However, this measurement is only carried out on a subset of frames. The coding of this result value can be found in section Coding of RX Lev.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	**:CALL:GSM:MSINfo:RXLSub?** The integer delivered back: **"49"**.

:CALL:GSM:MSINfo:SMS

Syntax	:CALL:GSM:MSINfo:SMS
Parameters	There are no parameters.
Description	This query will deliver back a boolean number indicating whether the mobile is able to handle short message service (SMS) or not. A **0** delivered back indicates that the mobile under test has no SMS capabilities.

Query	There is only a query form of this command available. The query will deliver back one boolean number.
Example	**:CALL:GSM:MSINfo:SMS?** The boolean number delivered back: **"1"**.

:CALL:GSM:MSINfo:ATTached?

Syntax	:CALL:GSM:MSINfo:ATTached?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	The query will deliver back one boolean number, indicating whether the mobile is GSM-attached to the base station simulated by the Willtek 3100. The GSM attach is usually part of the location update procedure. In a real network, the mobile can be paged only when it is attached and a TMSI (temporary mobile subscriber identity) is assigned to the mobile. Without the attach procedure, the mobile can be called (by its IMSI) if the Test SIM is being used.
Example	**:CALL:GSM:MSINfo:ATT?** The boolean number delivered back: **"1"**.

:CALL:GSM:MSINfo:EGSM

Syntax	:CALL:GSM:MSINfo:EGSM
Parameters	There are no parameters.
Description	This query will deliver back a boolean number indicating whether the mobile is able to use the extended GSM frequency range (channel numbers 0 and 975 to 1023) or not. A **0** indicates that the mobile under test cannot use the extended GSM frequency range.
Query	There is only a query form of this command available. The query will deliver back one boolean number.
Example	**:CALL:GSM:MSINfo:EGSM?** The boolean number delivered back: **"0"**.

:CALL:GSM:MSINfo:AFIVe

Syntax	:CALL:GSM:MSINfo:AFIVe
Parameters	There are no parameters.
Description	This query will deliver back the A5 ciphering key supported by the mobile.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	**:CALL:GSM:MSINfo:AFIVe?** The integer delivered back: **"3"**.

:CALL:GSM:MSINfo:TADVance

Syntax	:CALL:GSM:MSINfo:TADVance
Parameters	There are no parameters.
Description	This query will deliver back the timing advance currently used by the mobile (in bit periods). More details regarding the timing advance can be found in section Timeslots and Bursts.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	**:CALL:GSM:MSINfo:TADVance?** The integer delivered back: **"0"**.

:CALL:GSM:MSINfo:DUALband

Syntax	:CALL:GSM:MSINfo:DUALband
Parameters	There are no parameters.
Description	This query will deliver back an integer indicating whether the mobile is able to use both the GSM and PCN frequency ranges (GSM 900 and GSM 1800). A **0** indicates that the mobile under test is no dual-band mobile.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	**:CALL:GSM:MSINfo:DUALband?** The integer delivered back is: **"1"**.

:CALL:GSM:MSINfo:PCONtrol

Syntax	:CALL:GSM:MSINfo:PCONtrol
Parameters	There are no parameters.
Description	This query will deliver back the power control level currently used by the mobile. This figure is coded according to the scheme explained in section Power Control Levels and Tolerances.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	**:CALL:GSM:MSINfo:PCONtrol?** The integer delivered back is: **"16"**.

:CALL:GSM:MSINfo:EFRCapab

Syntax	`:CALL:GSM:MSINfo:EFRCapab`
Parameters	There are no parameters.
Description	This query will deliver back a boolean number indicating whether the mobile is capable of using enhanced full rate TCHs or not. A **0** means that the mobile under test is not able to support enhanced full rate TCHs.
Query	There is only a query form of this command available. The query will deliver back one boolean number.
Example	**:CALL:GSM:MSINfo:EFRCapab?** The boolean number delivered back is: **"1"**.

:CALL:WCDMa:POWer:ULTArget

Syntax	`:CALL:WCDMa:POWer:ULTArget <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value is `-50`, the maximum value `33`. The minimum resolution is `0.1`. The default value is `-50`.
Description	This command will set the UL target power.
Query	This query gets the UL target power.
Example	`:CALL:WCDM:POW:ULTA -10.0:CALL:WCDM:POW:ULTA?` Value returned in this example: `-10.0`

:CALL:WCDMa:POWer:ULTArget:FORCe

Syntax	`:CALL:WCDMa:POWer:ULTArget:FORCe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `NONe|MINimum|MAXimum`. Default is `NONe`.
Description	This command forces the UE to transmit with minimum or maximum power. If set to `NONe` the UE is set to transmit with the value specified by command: `CALL:WCDM:POW:ULTA`.
Query	The query form of this command returns the current setting for `PredefExp`.

Example	`:CALL:WCDM:POW:ULTA:FORC MAX`
	`:CALL:WCDM:POW:ULTA:FORC?`
	Value returned in this example: `MAX`.

:CALL:WCDMa:POWer:PREAmble

Syntax	`:CALL:WCDMa:POWer:PREAmble?`
Parameters	There are no parameters.
Description	There is no set form of this command.
Query	This query gets the calculated Preamble Initial Power.

Example	:CALL:WCDM:POW:PREA? Value returned in this example: -20.0

:CALL:WCDMa:ACTive

Syntax	:CALL:WCDMa:ACTive?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	This query checks if a call is established. Possible returns are NO|YES.
Example	:CALL:WCDM:ACT? The string delivered is, for example: "YES".

:CALL:WCDMa:NB:ORIGinate

Syntax	:CALL:WCDMa:NB:ORIGinate
Parameters	There are no parameters.
Description	Initiates a call setup by the base station (Willtek 3100).
Query	There is no query form of this command available.
Example	:CALL:WCDM:NB:ORIG This command will initiate a NB-originated call in the 3100 (i.e. the mobile phone will be called).

:CALL:WCDMa:NB:RELease

Syntax	:CALL:WCDMa:NB:RELease
Parameters	There are no parameters.
Description	This command will make the Willtek 3100 terminate the ongoing call.
Query	There is no query form of this command available.
Example	:CALL:WCDM:NB:REL This command will release the call.

:CALL:WCDMa:NB:LOOPback[:STARt]

Syntax	:CALL:WCDMa:NB:LOOPback[:STARt]
Parameters	There are no parameters.
Description	This command will set the UE into loopback test mode.
Query	This query checks if the UE is in loopback mode. Possible returns are ON|OFF.
Example	:CALL:WCDM:NB:LOOP:CALL:WCDM:NB:LOOP? Value returned in this example: "ON".

:CALL:WCDMa:NB:LOOPback:RELease

Syntax	:CALL:WCDMa:NB:LOOPback:RELease
Parameters	There are no parameters.
Description	This command will stop loopback mode.
Query	There is no query form of this command available.
Example	:CALL:WCDM:NB:LOOP:REL

:CALL:WCDMa:UE:NUMBer

Syntax	:CALL:WCDMa:UE:NUMBer?
Parameters	There are no parameters.
Description	This query will deliver the number dialed on the mobile as a string. In case the call was NB-originated, the string will be empty ("").
Query	There is only a query form of this command available. The query will deliver a string.
Example	:CALL:WCDM:UE:NUMB? The string delivered is, for example: "5551234".

:CALL:WCDMa:UE:IMEIsv

Syntax	:CALL:WCDMa:UE:IMEIsv?
Parameters	There are no parameters.
Description	This query will report the IMEI-(SV) of the UE.
Query	There is only a query form of this command available. The query will deliver a string.
Example	:CALL:WCDM:UE:IMEI? The string delivered is, for example: "5553434251234".

:CALL:WCDMa:UE:IMSI

Syntax	:CALL:WCDMa:UE:IMSI?
Parameters	There are no parameters.
Description	There is only a query form of this command available. The query will deliver a string.
Query	This query will deliver back a string containing the international mobile subscriber identity.
Example	:CALL:WCDM:UE:IMSI? The string delivered in this example could be: "001010123456789".

:CALL:WCDMa:UE:POWClass

Syntax	:CALL:WCDMa:UE:POWClass?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	This query will deliver the code representing the UE's power class. The query will deliver one integer.
Example	:CALL:WCDM:UE:POWC? The integer delivered back could be: 3.

:CALL:WCDMa:UE:MRAT

Syntax	:CALL:WCDMa:UE:MRAT?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	This query will report the supported Multi-RAT capabilities of the Mobile. Possible returns are NON|GSM.
Example	:CALL:WCDM:UE:MRAT? The string delivered in this example could be: "GSM".

:CALL:WCDMa:UE:MCAPabilities

Syntax	:CALL:WCDMa:UE:MCAPabilities?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	This query reports the UE's measurement capabilities for compressed mode for Uplink (first return value) and Downlink (second return value). 0 means compressed mode is not needed, 1 means compressed mode is needed.
Example	:CALL:WCDM:UE:MCAP? The string delivered in this example could be: 1,1.

:CALL:WCDMa:UE:POSitioning

Syntax	:CALL:WCDMa:UE:POSitioning?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	This query reports the supported positioning mechanisms. Positioning capabilities have 4 possible values: NONe|NETWork|GPS|BOTH (none, Network based, UE based, both). What is meant by UE based is UE GPS methods and what is meant by Network based is Network based methods.

Example	:CALL:WCDM:UE:POS? The string delivered in this example could be: "NET".

:CALL:WCDMa:UE:REGistration

Syntax	:CALL:WCDMa:UE:REGistration?
Parameters	There are no parameters.
Description	This command will check if the UE has registered. If not, it will set the Status Register Pending flag
Query	This query checks if the UE has registered with the Willtek 3100. Possible returns are NO|YES.
Example	:CALL:WCDM:UE:REG?:CALL:WCDM:REG

:CALL:WCDMa:UE:REGistration:CLEar

Syntax	:CALL:WCDMa:UE:REGistration:CLEar
Parameters	There are no parameters.
Description	This command causes the Willtek 3100 to erase any stored information about a previous registration by a mobile.
Query	There is no query form of this command available.
Example	:CALL:WCDM:UE:REG?:CALL:WCDM:REG

MEASure subsystem

The MEASure subsystem is probably the most important SCPI command subsystem of the 3100. There, you will find all commands required to acquire measurement results of the mobile under test.

:MEASure:GSM[:CONTinuous]:RFTX:GROup

Syntax	:MEASure:GSM[:CONTinuous]:RFTX:GROup
Parameters	There are no parameters.
Description	Starts a continuous measurement of the RF TX tests, specified with the :CONF:GSM:MEAS:GRO:RFTX command. To read out the latest measurement results, use the :FETCh:GSM:RFTX:GROup command. **Note:** Please keep in mind that the start of a new RF TX test will always terminate the preceding one instantly.
Query	The query form of this command will start the sequence of RF TX measurements as specified with the :CONF:GSM:MEAS:GRO:RFTX command. As soon as all measurements of the first sequence have been completed, a string will be delivered back, containing the related measurement result values, separated by commas. The type of the measurement result values (i.e. floating point real numbers, integers or boolean numbers) depends on the measurements specified. The order of the measurement result values within the string is as described below for the **:MEAS:GSM:CONT:RFTX:ALL** command.
Example	**:CONF:GSM:MEAS:GRO:RFTX POW,PRMS** **:MEASure:GSM:CONTinuous:RFTX:GROup?** In this example, the group of measurements is defined by a power level measurement combined with a RMS phase error measurement. The string returned is: **"4.53,9.98"**. Because of the internal order (see description of the command below), the first measurement result value delivered back is the RMS phase error, the second one the mobile's RF output power level.

:MEASure:GSM[:CONTinuous]:RFTX:ALL

Syntax

```
:MEASure:GSM[:CONTinuous]:RFTX:ALL
```

Parameters

There are no parameters.

Description

Starts a continuous measurement of the most important RF TX tests. To read out the latest measurement results, use the `:FETCh:GSM:RFTX:ALL` command.
The RF TX tests performed by this command are:
PPEAk, the peak phase error measurement,
PRMS, the root-mean square valued phase error measurement,
FREQuency means the frequency error of the mobile under test,
LENGth the length of the burst in microseconds,
UTIMe stands for the timing error of the mobile under test in microseconds (taking the ordered timing advance into account),
POWer is the mobile's RF output power level in dBm,
TEMPlate indicates whether the power/time template has been violated by a burst or not,
CORNer gives the RF power level in dBm at eight predefined positions of a burst (see section Corner Points for details),
FLATness gives an indication whether the mobile's RF output power is constant during the active part of the burst.
Notes:

- Please keep in mind that the start of a new RF TX test will always terminate the preceding one instantly.
- For a further description of the single measurements, see description of the related commands below.

Query

The query form of this command starts the measurements and - after all 19 measurements have been completed and all measurement results obtained - delivers a string, containing 19 measurement result values, separated with commas. The order and type of these measurement result values delivered back is as follows:
1. **PPEAk**, floating point real number, representing the result of the peak phase error measurement,
2. **PRMS**, floating point real number representing the result of the root-mean square valued phase error measurement,
3. **FREQuency**, floating point real number representing the mobile's frequency error,
4. **LENGth**, floating point real number representing the length of the burst in microseconds,
5. **UTIMe**, floating point real number representing the timing error of the mobile under test in microseconds (taking the ordered timing advance into account),
6. **POWer**, floating point real number representing the mobile's RF output power level in dBm,
7. **TEMPlate** boolean number. A **0** indicates that the power/time template (PTT) was **not** violated by the last burst measured, while a **1** means that the burst did violate the PTT.
8...15. **CORNer**, eight floating point real numbers, representing the RF power level in dBm at eight predefined positions of a burst.
16...19. **FLATness**, four floating point real numbers, representing the absolute minimum and maximum relative RF power levels of the mobile and their positions during the active part of the burst.
Note: For a further description of the single measurements, see description of the related commands below.

Example	**:MEASure:GSM:CONTinuous:RFTX:ALL?** In this case, all relevant RF TX measurements will be performed in a sequence. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing 19 measurement results: **"5.13,1.94,-2.22,557.0,0.1,11.22,0,-72.18, -61.91,-20.91,-0.05,-0.04,-17.97,-56.60, -73.95,-0.05,-0.01,0.02,-0.05"**.

:MEASure:GSM[:CONTinuous]:RFTX:PPEAk

Syntax	:MEASure:GSM[:CONTinuous]:RFTX:PPEAk
Parameters	There are no parameters.
Description	Starts a continuous measurement of the peak phase error. To read out the latest measurement result, use the :FETCh:GSM:RFTX:PPEAk command. For details regarding this RF TX measurement, refer to section Phase/Frequency Test Menu.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number with the physical dimension of degrees.
Example	**:MEASure:GSM:CONTinuous:RFTX:PPEAk?** The string returned is: **"5.13"**.

:MEASure:GSM[:CONTinuous]:RFTX:PRMS

Syntax	:MEASure:GSM[:CONTinuous]:RFTX:PRMS
Parameters	There are no parameters.
Description	Starts a continuous measurement of the root-mean square valued (RMS) phase error. To read out the latest measurement result, use the :FETCh:GSM:RFTX:PRMS command. For details regarding this RF TX measurement, refer to section Phase/Frequency Test Menu.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number with the physical dimension of degrees.
Example	**:MEASure:GSM:CONTinuous:RFTX:PRMS?** The string returned is: **"1.54"**.

:MEASure:GSM[:CONTinuous]:RFTX:FREQuency

Syntax	:MEASure:GSM[:CONTinuous]:RFTX:FREQuency
Parameters	There are no parameters.
Description	Starts a continuous measurement of the mobile's frequency error. To read out the latest measurement result, use the :FETCh:GSM:RFTX:FREQuency command. For details regarding this RF TX measurement, refer to section Phase/Frequency Test Menu.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number with the physical dimension of Hertz.
Example	**:MEASure:GSM:CONTinuous:RFTX:FREQuency?** The string returned is: **"-31.92"**.

:MEASure:GSM[:CONTinuous]:RFTX:LENGth

Syntax	:MEASure:GSM[:CONTinuous]:RFTX:LENGth
Parameters	There are no parameters.
Description	Starts a continuous measurement of the burst length. To read out the latest measurement result, use the :FETCh:GSM:RFTX:LENGth command. For details regarding this RF TX measurement, refer to section Burst Test Menu.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number with the physical dimension of microseconds.
Example	**:MEASure:CONTinuous:RFTX:LENGth?** The string returned is: **"557.0"**.

:MEASure:GSM[:CONTinuous]:RFTX:UTIMe

Syntax	:MEASure:GSM[:CONTinuous]:RFTX:UTIMe
Parameters	There are no parameters.
Description	Starts a continuous measurement of the uplink timing error. To read out the latest measurement result, use the :FETCh:GSM:RFTX:UTIMe command. For details regarding this RF TX measurement, refer to section Burst Results Display.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number with the physical dimension of microseconds.
Example	**:MEASure:GSM:CONTinuous:RFTX:UTIMe?** The string returned is: **"0.1"**.

:MEASure[:CONTinuous]:RFTX:POWer

Syntax	:MEASure[:CONTinuous]:RFTX:POWer
Parameters	There are no parameters.
Description	Starts a continuous measurement of the mobile's RF output power level. To read out the latest measurement result, use the :FETCh:GSM:RFTX:POWer command. For details regarding this RF TX measurement, refer to section Burst Results Display.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number with the physical dimension of dBm.
Example	**:MEASure:GSM:CONTinuous:RFTX:POWer?** The string returned is: **"11.22"**.

:MEASure:GSM[:CONT]:RFTX:TEMPlate

Syntax	:MEASure:GSM[:CONT]:RFTX:TEMPlate
Parameters	There are no parameters.
Description	Starts a continuous check of the UL burst against the power/time template (PTT). To read out the latest measurement result, use the :FETCh:GSM:RFTX:TEMPlate command. For details regarding this RF TX measurement, refer to section Burst Test Menu. The power/time template (PTT) is user-definable. For details, please refer to section Power/Time Templates (PTT). **Note:** The power/time template can also be set using the :CALC:RFTX:TEMP commands.
Query	The query form of this command will start the (continuous) check and return a result value as soon as the first check has been completed. The string delivered back will contain one boolean number. A **0** indicates that there was **no** violation of the PTT, while a **1** means that the last burst measured did violate the PTT.
Example	**:MEASure:GSM:CONTinuous:RFTX:TEMPlate** This command will start the continuous check of the burst against the PTT.

:MEASure:GSM[:CONT]:RFTX:CORNer

Syntax	:MEASure:GSM[:CONT]:RFTX:CORNer
Parameters	There are no parameters.
Description	Starts a continuous measurement of the mobile's RF output power level on eight user-definable positions of the ramping phases of a burst (the so-called corner points). To read out the latest measurement results, use the :FETCh:GSM:RFTX:CORNer command. For details regarding this RF TX measurement, please refer to section Burst Results Display. More details regarding the corner points can be found in section Corner Points. **Note:** The corner points can also be set using the :CALC:RFTX:CORN commands.

Query	The query form of this command will start the measurement and return eight measurement result values as soon as the first measurement has been completed. The string delivered back will contain eight floating point real numbers with the physical dimension of dBm. The single values are separated by commas.
Example	**`:MEASure:GSM:CONTinuous:RFTX:CORNer?`** The string returned is: **`"-72.18,-61.91,-20.91,-0.05, -0.04,-17.97,-56.60,-73.95"`**.

`:MEASure:GSM[:CONT]:RFTX:FLATness`

Syntax	`:MEASure:GSM[:CONT]:RFTX:FLATness`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the flatness of the mobile's burst during the active part of it. In short words, it is a search for both the absolute minimum and the absolute maximum of the mobile's output power level during the active part of the burst. This measurement will collect four measurement result values: – the first value is the minimum RF power level in dB, – the second one is the position of this minimum within the burst in bit positions, – the third value is the maximum RF power level in dB, while – the last value is the position of the maximum within the burst in terms of bit position. The resolution of this measurement in the time domain is approx. 0.825 microseconds, equaling approx. 1/5 bit periods. To read out the latest measurement result, use the `:FETCh:GSM:RFTX:FLATness` command.
Query	The query form of this command will start the measurement and return four measurement result values as soon as the first measurement has been completed. The string delivered back will contain four floating point real numbers with the physical dimensions explained above. The single values are separated by commas.
Example	**`:MEASure:GSM:CONTinuous:RFTX:FLATness?`** The string returned is: **`"-0.12,113,7,0.56,34.0"`**.

`:MEASure:GSM[:CONT]:RFTX:FPOWer`

Syntax	`:MEASure:GSM[:CONT]:RFTX:FPOWer`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the fast burst peak power level. This measurement is very similar to the **`:MEAS:GSM:RFTX:POWer`** measurement, but it is faster. To read out the latest measurement result, use the `:FETCh:GSM:RFTX:FPOWer` command. For further details, refer to the description of the `:CONF:GSM:MEAS:LEV:EXP` command.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number with the physical dimensions of dBm.
Example	**`:MEASure:GSM:CONTinuous:RFTX:FPOWer?`** The string returned is: **`"13.05"`**.

:MEASure:GSM[:CONTinuous]:BLOCkdata:BURStshape

Syntax	`:MEASure:GSM[:CONTinuous]:BLOCkdata:BURStshape`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the mobile's UL burst shape. To read out the measurement result values of the last burst sampled, use the `:FETCh:GSM:RFTX:BLOCkdata:BURStshape` command. The aim of this measurement is to obtain the pure RF power levels, sampled by the Willtek 3100 in order to display the shape of the burst graphically on another computer or to perform some special analysis of the burst shape. For details regarding the burst measurement, also refer to section Burst Test Menu.
Query	The query form of this command will start the measurement and return an array of measurement result values as soon as the first measurement has been completed. The string delivered back will contain 711 floating point real numbers with the physical dimension of dB (except the first two). The first data position of the array is the absolute position of the middle of the burst (i.e. bit 73) within the burst data supplied. In order to allocate the RF power level obtained for the middle of the burst, **add 2** to this first value of the array to find the relative position within this data array (as the first two positions of this array are no data values). The second number is the **absolute** peak power level of the burst in dBm, obtained at the middle of the burst. All following 709 data values are **relative** RF power levels (in dB) with the peak power level (second data value of the array) being the reference. The single measurement result values are separated by commas.
Example	**`:MEAS:GSM:CONT:BLOC:BURStshape?`** The array returned is: **`"353.0,10.0,-73.5,-69.6,-70.8, ...-0.1,0.0,0.2,...-68.4"`**. The data have the following meaning: **`353.0`** on the first position means that the absolute position of the middle of the burst is position 353 of the burst data array. Thus, it can be found on position 355 of the data block returned. The relative power level at the middle of the burst will always be **`0.0`** as it is the reference point. The peak power of the burst (obtained at the reference point 'middle of the burst') can be found on the second position of the data array and is **`10.0`** dBm. Then follows the relative RF power levels sampled at the various time positions of the burst (**`-73.5,-69.6,...`** dB). This means that the absolute power level at the first position sampled is -63,5 dBm (-73.5+10.0). All the remaining data values are relative RF power levels sampled by the Willtek 3100 during the burst period.

:MEASure:GSM[:CONTinuous]:BLOCkdata:PHASeerror

Syntax	`:MEASure:GSM[:CONTinuous]:BLOCkdata:PHASeerror`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the mobile's phase error curve. To read out the measurement result values of the latest phase error curve, use the `:FETCh:GSM:RFTX:BLOCkdata:PHASeerror` command. The aim of this measurement is to obtain the pure phase error levels, calculated by the Willtek 3100 in order to graphically display the phase error curve on another computer or to perform some special analysis of the phase error on an external computer. For details regarding the phase error measurement, please also refer to section Phase/ Frequency Test Menu.
Query	The query form of this command will start the measurement and return an array of measurement result values as soon as the first measurement has been completed. The string delivered back will contain 711 floating point real numbers with the physical dimensions of degrees (except the first two). The first number is the absolute position of the middle of the burst (i.e. bit 73). In order to allocate the phase error obtained for bit 73, **add 2** to this first value of the array to find the relative position within this data array (as the first two positions of this array are no data values). The second number of this data array is always set to **0.0**. All following 709 data values are phase errors, forming the curve of the phase error during a burst. The single measurement result values are separated by commas.
Example	**:MEAS:GSM:CONT:BLOCkdata:PHASeerror?** The array returned is: **"353.0,0.0,0.0,0.21,0.41,0.32, ...-2.17,-2.01,-1.87,...0.0"**. The data have the following meaning: **353.0** on the first position means that the absolute position of the middle of the burst is position 353 of the pure data array. Thus, it can be found on position 355 of the entire data block. The result value **0.0** on the second position has no particular meaning. However, it may be used to detect a phase error data array. All the remaining data levels are phase errors calculated, forming the graph of the phase error.

:MEASure[:CONTinuous]
:BLOCkdata:AFSPectrum[:CURRent]

Syntax	`:MEASure[:CONTinuous]:BLOCkdata:AFSPectrum[:CURRent]`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the audio spectrum curve. To read out the measurement result values of the latest audio spectrum curve, use the `:FETCh:GSM:RFTX:BLOCkdata:AFSPectrum` command. The aim of this measurement is to obtain the pure audio spectrum levels, calculated by the Willtek 3100 in order to graphically display the audio spectrum curve on another computer or to perform some special analysis of the audio spectrum. For details regarding the audio spectrum measurement, also refer to the short-form manual, AF options. **Note:** Any audio measurements require the Audio Option to be installed on your Willtek 3100.

Query	The query form of this command will start the measurement and return an array of measurement result values as soon as the first measurement has been completed. The string delivered back will contain a set of floating point real numbers with the physical dimension of dB. The number of measurement result values depends on the span and the resolution currently set for the audio spectrum analyzer. The data array returned will only contain measurement result data values (there are no offset or reference levels in the first data positions). The single measurement result values are separated by commas.
Example	**`:MEAS:CONT:BLOCkdata:AFSPectrum:CURRent?`** The array returned is: **`"-120.0,-119.5,-119.7, ...-30.4,-30.4,-30.3,...-119.7"`**.

`:MEASure[:CONTinuous]:BLOCkdata:AFSPectrum:AVG`

Syntax	`:MEASure[:CONTinuous]:BLOCkdata:AFSPectrum:AVG <int1>`
Parameters	**`int1`** is an integer. The minimum value for **`int1`** is **0**, the maximum value is **100**. The default value for **`int1`** is **0**.
Description	Starts a process that measures and averages the audio spectrum for a specified number of times (**`int1`**). The aim of this measurement is to obtain a 'smoothed' audio spectrum curve that can then be displayed on another computer or used to perform some special analysis of the audio spectrum. For details regarding the audio spectrum measurement, also refer to the short-form manual, AF options. **Note:** Any audio measurements require the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command will start the averaging process and return an array of measurement result values as soon as all the measurements have been completed. The string delivered back will contain a set of floating point real numbers with the physical dimension of dB. The number of measurement result values depends on the span and the resolution currently set for the audio spectrum analyzer. The data array returned will only contain measurement result data values (there are no offset or reference levels in the first data positions). The single measurement result values are separated by commas.
Example	**`:MEAS:CONT:BLOCkdata:AFSPectrum:AVG? 25`** The array returned is: **`"-61.0,-61.0,-61.1, ...0.0,0.0,0.0,...-61.0"`**.

:MEASure:GSM[:CONTinuous]
:BLOCkdata:MSPectrum[:CURRent]

Syntax	:MEASure:GSM[:CONTinuous]:BLOCkdata:MSPectrum[:CURRent]
Parameters	There are no parameters.
Description	Starts a continuous measurement of the modulation spectrum curve. To read out the measurement result values of the latest modulation spectrum curve, use the :FETCh:GSM:RFTX:BLOCkdata:MSPectrum command. The aim of this measurement is to obtain the pure modulation spectrum levels, calculated by the Willtek 3100 in order to graphically display the modulation spectrum curve on another computer or to perform some special analysis of the modulation spectrum.
Query	The query form of this command will start the measurement and return an array of measurement result values as soon as the first measurement has been completed. The string delivered back will contain a set of floating point real numbers with the physical dimension of dB. The number of measurement result values depends on the span and the resolution currently set for the modulation spectrum analyzer. The span can be set using the :RFSP:MSP:SPAN command; the resolution may be set using the :RFSP:MSP:RES command. The data array returned will only contain measurement result data values (there are no offset or reference levels in the first data positions). The single measurement result values are separated by commas.
Example	**:MEAS:GSM:CONT:BLOCkdata:MSPectrum?** The array returned is:**"-62.0,-61.5,-62.2, ...-0.4,0.0,-0.3,...-62.4"**.

:MEASure:GSM[:CONTinuous]
:BLOCkdata:MSPectrum:AVG

Syntax	:MEASure:GSM[:CONTinuous]:BLOCkdata:MSPectrum:AVG <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Starts a process that measures and averages the modulation spectrum for a specified number of times. The aim of this measurement is to obtain a 'smoothed' modulation spectrum curve in order to graphically display the modulation spectrum on another computer or to perform some special analysis. **Note:** To read out the measurement result values of the averaging process, the **:FETCh:GSM:RFTX:BLOCkdata:MSPectrum** command may be used.

Query	The query form of this command will start the averaging process and return an array of measurement result values as soon as all the measurements have been completed. The string delivered back will contain a set of floating point real numbers with the physical dimension of dB. The number of measurement result values depends on the span and the resolution currently set for the modulation spectrum analyzer. The span can be set using the **:RFSP:MSP:SPAN** command; the resolution may be set using the **:RFSP:MSP:RES** command. The data array returned will only contain measurement result data values (there are no offset or reference levels in the first data positions). The single measurement result values are separated by commas.
Example	**:MEAS:GSM:CONT:BLOCkdata:MSPectrum:AVG? 25** The array returned is: **"-61.0,-61.0,-61.1, ...0.0,0.0,0.0,...-61.0"**.

:MEASure:GSM[:CONTinuous]
:BLOCkdata:PSCShape[:CURRent]

Syntax	:MEASure[:GSM][:CONTinuous]:BLOCkdata:PSCShape[:CURRent]
Parameters	There are no parameters.
Description	Starts a continuous measurement of the current shape curve. To read out the measurement result values of the latest curve, use the :FETCh:BLOCkdata:PSCShape command. The aim of this measurement is to obtain the current consumption values at each point in time during an entire TDMA frame. The time/current points are also drawn as a curve on the Power Supply graphical display.
Query	The query form of this command will start the measurement and return an array of measurement result values as soon as the measurement has been completed. The string delivered back will contain 886 floating point real numbers with the physical dimension of mA. The current is sampled at a rate of 192 kHz, starting with the beginning of a new downlink TDMA frame.
Example	**:MEAS:GSM:CONT:BLOCkdata:PSCShape?** Returns 886 measurement values, describing the current shape over a TDMA frame.

:MEASure:GSM[:CONT]:RFRX:RBER:ALL

Syntax	`:MEASure:GSM[:CONT]:RFRX:RBER:ALL`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the residual bit error ratios (RBER). To read out the latest measurement results, use the `:FETCh:GSM:RFRX:RBER:ALL` command. The RF RX tests performed by this command are: **CIB**, the residual bit error ratio of the class Ib bits, **CII**, the residual bit error ratio of the class II bits, and **FER**, the frame erasure ratio. **Notes:** – More details regarding RBER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – Please keep in mind that the start of a new RF RX test will always terminate the preceding one instantly. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command starts the measurements and - after all measurements have been completed and all measurement results obtained - delivers a string, containing three floating point real numbers. The order and meaning of these numbers delivered back is as follows: 1. **CIB**, representing the residual bit error ratio of the class Ib bits, 2. **CII**, representing the residual bit error ratio of the class II bits and 3. **FER**, representing the frame erasure ratio. The single result values are separated by commas.
Example	**:MEAS:GSM:CONT:RFRX:RBER:ALL?** In this case, all RBER measurements will be performed. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing the three measurement result values: **"0.0,1.2,0.0"**.

:MEASure:GSM[:CONT]:RFRX:RBER:CIB

Syntax	`:MEASure:GSM[:CONT]:RFRX:RBER:CIB`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the residual bit error ratio of the class Ib bits. To read out the latest measurement result, use the `:FETCh:GSM:RFRX:RBER:CIB` command. **Notes:** – More details regarding RBER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number, representing the bit error ratio.

Example	**`:MEASure:CONTinuous:RFRX:RBER:CIB?`** Measurement result string delivered back: **`"0.1"`**.

`:MEASure:GSM[:CONT]:RFRX:RBER:CII`

Syntax	`:MEASure:GSM[:CONT]:RFRX:RBER:CII`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the residual bit error ratio on the class II bits. To read out the latest measurement result, use the `:FETCh:GSM:RFRX:RBER:CII` command. **Notes:** – More details regarding RBER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number, representing the bit error ratio.
Example	**`:MEAS:GSM:CONT:RFRX:RBER:CII?`** Measurement result string delivered back: **`"1.2"`**.

`:MEASure:GSM[:CONT]:RFRX:RBER:FER`

Syntax	`:MEASure:GSM[:CONT]:RFRX:RBER:FER`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the frame erasure ratio. To read out the latest measurement result, use the `:FETCh:GSM:RFRX:RBER:FER` command. **Notes:** – More details regarding RBER testing can be found in section BER Live Display. – The number of samples to be used for testing can be set, using the **`:CONF:GSM:BER:COUNt`** command.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number, representing the frame erasure ratio.
Example	**`:MEAS:GSM:CONT:RFRX:RBER:FER?`** Measurement result string delivered back: **`"0.0"`**.

:MEASure:GSM[:CONT]:RFRX:BER:ALL

Syntax	`:MEASure:GSM[:CONT]:RFRX:BER:ALL`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the (nonresidual) bit error ratios (BER). To read out the latest measurement results, use the `:FETCh:GSM:RFRX:BER:ALL` command. The RF RX tests performed on this command are: **CIA**, the bit error ratio of the class Ia bits, **CIB**, the bit error ratio of the class Ib bits, and **CII**, the bit error ratio of the class II bits. **Notes:** – More details regarding BER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – Please keep in mind that the start of a new RF RX test will always terminate the preceding one instantly. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command starts the measurements and - after all measurements have been completed and all measurement results obtained - delivers a string, containing three floating point real numbers. The order and meaning of these numbers delivered back is as follows: 1. **CIA**, representing the bit error ratio of the class Ia bits, 2. **CIB**, representing the bit error ratio of the class Ib bits, and 3. **CII**, representing the bit error ratio of the class II bits. The single measurement result values are separated by commas.
Example	**:MEASure:GSM:CONTinuous:RFRX:BER:ALL?** In this case, all BER measurements will be performed in a sequence. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing the three measurement result values: **"0.0,0.0,0.2"**.

:MEASure:GSM[:CONT]:RFRX:BER:CIA

Syntax	`:MEASure:GSM[:CONT]:RFRX:BER:CIA`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the (nonresidual) bit error ratio of the class Ia bits. To read out the latest measurement result, use the `:FETCh:GSM:RFRX:BER:CIA` command. **Notes:** – More details regarding BER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number, representing the bit error ratio.

Example	**`:MEAS:GSM:CONT:RFRX:BER:CIA?`** Measurement result string delivered back: **`"0.0"`**.

`:MEASure:GSM[:CONT]:RFRX:BER:CIB`

Syntax	`:MEASure:GSM[:CONT]:RFRX:BER:CIB`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the (nonresidual) bit error ratio of the class Ib bits. To read out the latest measurement result, use the `:FETCh:GSM:RFRX:BER:CIB` command. **Notes:** – More details regarding BER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number, representing the bit error ratio.
Example	**`:MEAS:GSM:CONT:RFRX:BER:CIB?`** Measurement result string delivered back: **`"0.1"`**.

`:MEASure:GSM[:CONT]:RFRX:BER:CII`

Syntax	`:MEASure:GSM[:CONT]:RFRX:BER:CII`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the (nonresidual) bit error ratio on the class II bits. To read out the latest measurement result, use the `:FETCh:GSM:RFRX:BER:CII` command. **Notes:** – More details regarding BER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number, representing the bit error ratio.
Example	**`:MEAS:GSM:CONT:RFRX:BER:CII?`** Measurement result string delivered back: **`"0.2"`**.

:MEASure:GSM[:CONTinuous]:RFRX:FBER

Syntax	`:MEASure:GSM[:CONTinuous]:RFRX:FBER`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the fast bit error ratio (C-type loop). To read out the latest measurement result, use the `:FETCh:GSM:RFRX:FBER` command. **Notes:** – More details regarding Fast BER testing can be found in section BER Live Display. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one floating point real number, representing the bit error ratio.
Example	**`:MEAS:GSM:CONT:RFRX:FBER?`** Measurement result string delivered back: **`"1.5"`**.

:MEASure:GSM[:CONT]:RFRX:RTDelay

Syntax	`:MEASure:GSM[:CONT]:RFRX:RTDelay`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the round trip delay for data during bit error ratio measurements. To read out the latest measurement result, use the `:FETCh:GSM:RFRX:RTD` command. **Note:** More details regarding the round-trip delay can be found in section BER Live Display.
Query	The query form of this command will start the measurement and return a measurement result value as soon as the first measurement has been completed. The string delivered back will contain one integer, representing the round trip delay.
Example	**`:MEAS:GSM:CONT:RFRX:RTDelay?`** Measurement result string delivered back: **`"8"`**.

:MEAS:GSM[:CONT]:RFSP:ACPM:MODulation

Syntax	`:MEAS:GSM[:CONT]:RFSP:ACPM:MODulation`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the ACPM modulation spectrum. To read out the latest measurement results, use the `:FETCh:GSM:RFSP:ACPM:MOD` command. **Note:** According to GSM specification 05.05, this measurement will be performed on (±)100 kHz and (±)250 kHz offset from the carrier as well as on all multiples of 200 kHz offset from the carrier up to (and including) an offset of (±)1.8 MHz. If one includes the carrier frequency as well (0.0 dBc), this adds up to the 23 measurement values returned. All measurements will be performed with a measuring bandwidth of 30 kHz.

Query	The query form of this command starts the measurement and delivers a string, containing 23 measurement result values separated by commas. These 23 measurement result values are floating point real numbers with the physical dimension dBc (dB relative to the carrier, being the transmission frequency of the mobile). The 23 values represent the power in a 30 kHz bandwidth at the following frequencies: -1.8 MHz, -1.6 MHz, -1.4 MHz, -1.2 MHz, -1.0 MHz, -800 kHz, -600 kHz, -400 kHz, -250 kHz, -200 kHz, -100 kHz, 0 kHz, +100 kHz, 200 kHz, 250 kHz, 400 kHz, 600 kHz, 800 kHz, 1.0 MHz, 1.2 MHz, 1.4 MHz, 1.6 MHz, 1.8 MHz.
Example	**`:MEAS:GSM:CONT:RFSP:ACPM:MODulation?`** In this case, a measurement of the modulation spectrum will be performed. As soon as the measurement has been completed, a string will be delivered back containing 23 floating point real numbers.

`:MEASure:GSM[:CONTinuous]:RFSPectrum:ACPM:MODPower?`

Syntax	`:MEASure:GSM[:CONTinuous]:RFSPectrum:ACPM:MODPower?`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the ACPM modulation spectrum. To read out the latest measurement results, use the `:FETCh:GSM:RFSP:ACPM:MODP` command. Notes: – The first result is the absolute power in dBm, measured at the center frequency. The remaining 23 values are the same as in the `ACPM:MODulation` command. – According to GSM specification 05.05, this measurement will be performed on (±)100 kHz and (±)250 kHz offset from the carrier as well as on all multiples of 200 kHz offset from the carrier up to (and including) an offset of (±)1.8 MHz. Including the carrier frequency (0.0 dBc), this adds up to the 23 measurement values returned. All measurements will be performed with a measuring bandwidth of 30 kHz.
Query	The query form of this command starts the measurement and delivers back a string, containing the absolute power at the center frequency (within a 30 kHz bandwidth) as the first value with the physical dimension of dBm and after that 23 measurement result values, all separated by commas. These 23 measurement result values are floating point real numbers with the physical dimension dBc (dB relative to the carrier, being the transmission frequency of the mobile).
Example	`:MEAS:RFSP:ACPM:MODP?` In this case, a measurement of the modulation spectrum will be performed. As soon as the measurement has been completed, a string will be delivered back containing 24 floating point real numbers.

:MEAS:GSM[:CONT]:RFSP:ACPM:TRANsient

Syntax	:MEAS:GSM[:CONT]:RFSP:ACPM:TRANsient
Parameters	There are no parameters.
Description	Starts a continuous measurement of the ACPM spectrum due to switching transients. To read out the latest measurement results, use the :FETCh:GSM:RFSP:ACPM:TRANS command. **Note:** According to GSM specification 05.05, this measurement will be performed on 0, (±)400, (±)600, (±)1.200 and (±)1.800 kHz offset from the carrier. All measurements will be performed with a measuring bandwidth of 30 kHz. For additional information, also the measurement values of the 30 kHz windows below and above the frequencies specified by GSM standard 05.05 will be delivered back. This adds up to 9 x 3 = 27 measurement result values, that are returned.
Query	The query form of this command starts the measurement and delivers a string, containing 27 measurement result values separated by commas. These 27 measurement result values are floating point real numbers with the physical dimension dBm.
Example	**:MEAS:GSM:CONT:RFSP:ACPM:TRANsient?** In this case, a measurement of the modulation spectrum will be performed. As soon as the measurement has been completed, a string will be delivered back containing 27 floating point real numbers.

:MEASure[:CONT]:AFANalyser:GROup

Syntax	:MEASure[:CONT]:AFANalyser:GROup
Parameters	There are no parameters.
Description	Starts a continuous measurement of the audio tests, specified with the :CONF:MEAS:GRO:AFAN command. To read out the latest measurement results, use the :FETCh:AFAN:GROup command. **Notes:** – Please keep in mind that the start of a new AF test will always terminate all other measurements. – To perform any audio measurements on your Willtek 3100, the Audio Option must be installed.
Query	The query form of this command will start the measurements and deliver back a string, containing the latest set of measurement result values. All measurement result values returned will be floating point real numbers. The order of the measurement result values returned is as described below (see command **:MEAS:AFAN:ALL**). The single measurement result values are separated by commas.
Example	**:CONF:MEAS:GRO:AFAN SIN,FREQ** **:MEASure:CONTinuous:AFANalyser:GROup?** In this example, the group of measurements is defined by a SINAD measurement, combined with an AF frequency measurement. The measurement result string returned is: **"1000.0,50.5"**. Because of the internal order, the first measurement result value delivered back is the audio frequency, the second one the SINAD.

:MEASure[:CONT]:AFANalyser:ALL

Syntax `:MEASure[:CONT]:AFANalyser:ALL`

Parameters There are no parameters.

Description Starts a continuous measurement of the most important audio tests. To read out the latest measurement results, use the `:FETCh:AFAN:ALL` command.
The audio tests performed by this command are:
ACVPeakp, the peak-to-peak measurement of an AC voltage,
ACVRms, the RMS-valued measurement of an AC voltage,
DCV is used to measure AC ripple on DC lines (this measurement gives the root-mean spare voltage of the AC component of the applied DC signal).
FREQuency is the measurement of the audio frequency in Hertz,
DISTortion is the distortion measurement on the third harmonic of a sine wave and expressed in %, while
SINad is the measurement of the signal to noise ratio, expressed in dB.
Notes:
- Please keep in mind that the start of a new AF test will always terminate all other measurements.
- Any AF measurement will need the Audio Option to be installed on your Willtek 3100.

Query The query form of this command starts the measurements and – after all measurements have been completed and all measurement results obtained – delivers a string, containing six floating point real numbers, representing the six measurement result values. The order of these measurement result values delivered back is as follows:
1. **ACVPeakp**, representing the AC peak-to-peak voltage of the AF signal,
2. **ACVRms**, representing the RMS-valued AC voltage of the AF signal,
3. **DCV**, representing the RMS-valued AC voltage on an applied DC signal,
4. **FREQuency**, representing the audio frequency,
5. **DISTortion**, representing the third-harmonic distortion of the applied sine-wave AF signal and,
6. **SINad**, representing the signal to noise ratio of the applied AF signal.

Example **:MEASure:CONTinuous:AFANalyser:ALL?**
In this case, all audio measurements will be performed in a sequence. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing six measurement result values:
"0.7,0.25,0.0,1000.0,0.3,50.5".

:MEAS[:CONT]:AFAN:ACVoltage:PEAKp

Syntax `:MEAS[:CONT]:AFAN:ACVoltage:PEAKp`

Parameters There are no parameters.

Description Starts a continuous measurement of the AC peak-to peak voltage of the AF signal applied to the audio analyzer. To read out the latest measurement result, use the `:FETCh:AFAN:ACV:PEAK` command.
Note: Any AF test command needs the Audio Option to be installed on your Willtek 3100.

Query The query form of this command starts the measurement and delivers a string, containing a floating point real number, representing the first measurement result value. The physical dimension is volt, measured peak-to-peak (V_{pp}).

Example	**:MEAS:CONT:AFAN:ACV:PEAKp?** String delivered back: **"0.7"**.

:MEAS[:CONT]:AFAN:ACVoltage:RMS

Syntax	:MEAS[:CONT]:AFAN:ACVoltage:RMS
Parameters	There are no parameters.
Description	Starts a continuous measurement of the RMS-valued AC voltage of the AF signal applied to the audio analyzer. To read out the latest measurement result, use the :FETCh:AFAN:ACV:RMS command. **Note:** Any AF test command needs the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command starts the measurement and delivers a string, containing a floating point real number, representing the first measurement result value. The physical dimension is volt, RMS-valued (V_{rms}).
Example	**:MEAS:CONT:AFAN:ACVoltage:RMS?** String delivered back: **"0.25"**.

:MEAS][:CONT]:AFAN:DCVoltage

Syntax	:MEAS][:CONT]:AFAN:DCVoltage
Parameters	There are no parameters.
Description	Starts a continuous measurement of the RMS-valued AC component of an applied DC signal. To read out the latest measurement result, use the :FETCh:AFAN:DCV command. **Note:** Any AF test command needs the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command starts the measurement and delivers a string, containing a floating point real number, representing the first measurement result value. The physical dimension is volt, RMS-valued (V_{rms}).
Example	**:MEAS:CONT:AFAN:DCVoltage?** String delivered back: **"0.025"**.

:MEAS[:CONT]:AFAN:FREQuency

Syntax	:MEAS[:CONT]:AFAN:FREQuency
Parameters	There are no parameters.
Description	Starts a continuous measurement of the frequency of the audio signal applied to the audio analyzer. To read out the latest measurement result, use the :FETCh:AFAN:FREQ command. **Note:** Any AF test command needs the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command starts the measurement and delivers a string, containing a floating point real number, representing the first measurement result value. The physical dimension is Hertz (Hz).

Example	**:MEAS:CONT:AFAN:FREQuency?** String delivered back: **"1000.0"**.

:MEAS[:CONT]:AFAN:DISTortion

Syntax	:MEAS[:CONT]:AFAN:DISTortion
Parameters	There are no parameters.
Description	Starts a continuous measurement of the third-harmonic distortion of a sine wave applied to the audio analyzer. To read out the latest measurement result, use the :FETCh:AFAN:DIST command. **Note:** Any AF test command needs the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command starts the measurement and delivers a string, containing a floating point real number, representing the first measurement result value. The physical dimension is percent (%).
Example	**:MEAS:CONT:AFAN:DISTortion?** String delivered back: **"0.3"**.

:MEASure[:CONT]:AFANalyser:SINad

Syntax	:MEASure[:CONT]:AFANalyser:SINad
Parameters	There are no parameters.
Description	Starts a continuous measurement of the SINAD (i.e. signal to noise ratio). To read out the latest measurement result, use the :FETCh:AFAN:SIN command. **Note:** Any AF test command needs the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command starts the measurement and delivers a string, containing one floating point real number, representing the first measurement result value. The physical dimension is dB.
Example	**:MEAS:CONT:AFANalyser:SINad?** Value delivered back: **"50.5"**.

:MEASure[:CONTinuous]:PSUPply:GROup

Syntax	:MEASure[:CONTinuous]:PSUPply:GROup
Parameters	There are no parameters.
Description	Starts a continuous measurement of the power supply tests specified with the :CONF:MEAS:GRO:PSUP command. To read out the latest measurement results, use the :FETCh:PSUP:GROup command. **Notes** – To perform any power and current consumption measurements on your Willtek 3100, the MS Power Supply and Current Measurements Option must be installed. – For further details regarding group measurements, please refer to section :MEASure:...:GROup.

Query	The query form of this command will start the measurements and deliver back a string containing the latest set of measurement result values. All measurement result values returned will be floating point real numbers. The order of the measurement result values returned is as described below (see command `:MEAS:PSUP:ALL`). The single measurement result values are separated by commas.

Example	`:CONF:MEAS:GRO:PSUP ACUR,APOW`
	`:MEAS:PSUP:GRO?`
	In this example, the group of measurements is defined by a power consumption measurement combined with a current consumption measurement. The measurement result string returned in this example is: `"863.6,304.2"`.
	Because of the internal order, the first measurement result value delivered back is the power consumption, the second one the average current consumption measurement.

:MEASure[:CONTinuous]:PSUPply:ALL

Syntax	`:MEASure[:CONTinuous]:PSUPply:ALL`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the power and current consumption tests. To read out the latest measurement results, use the `:FETCh:PSUP:ALL?` query. The power supply/current measurement tests performed by this command are: `APOW` for the average power consumption measurement, measured in mW; `ACUR` for the average current consumption measurement in mA, `PCUR` for the peak current consumption in mA. **Notes** – Any power and current consumption measurement will need the MS Power Supply and the Current Measurement Options to be installed on your Willtek 3100. – For further details regarding group measurements, please refer to section `:MEASure:...:GROup`.
Query	The query form of this command starts the measurements and - after all measurements have been completed and all measurement results obtained - delivers back a string containing three floating point real numbers, representing the three measurement result values. The order of these measurement result values delivered back is as follows: – `APOW`, representing the average power consumption in mW, – `ACUR`, representing the average current consumption in mA, – `PCUR`, representing the peak current consumption in mA.
Example	`:MEAS:PSUP:ALL?` In this case, all power/current consumption measurements are performed in one go. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing three measurement result values, for example `"863.6,304.2,1352.9"`.

:MEASure[:CONTinuous]:PSUPply:APOWer

Syntax	`:MEASure[:CONTinuous]:PSUPply:APOWer`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the power consumption test. To read out the latest measurement result, use the `:FETCh:PSUP:APOW?` query. **Note:** Any power and current consumption measurement will need the MS Power Supply and the Current Measurement Options to be installed on your Willtek 3100.
Query	The query form of this command starts the measurements and - after all measurements have been completed and all measurement results obtained - delivers back a string containing the measurement result value. The value represents the average power consumption in mW.
Example	`:MEAS:PSUP:APOW?` In this case, all the average power consumption measurements is performed. As soon as the measurement has been completed, a string will be delivered back containing the measurement result value, for example `"863.6"`.

:MEASure[:CONTinuous]:PSUPply:ACURrent

Syntax	`:MEASure[:CONTinuous]:PSUPply:ACURrent`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the average current consumption test. To read out the latest measurement result, use the `:FETCh:PSUP:ACUR` command. Any power and current consumption measurement will need the MS Power Supply and the Current Measurement Options to be installed on your Willtek 3100.
Query	The query form of this command starts the measurement and - after the measurement has been completed and the measurement result obtained - delivers back a string containing a floating point real number representing the average current consumption in mA.
Example	`:MEAS:PSUP:ACUR?` In this case, the average current consumption measurement is performed. As soon as the measurement has been completed, a string will be delivered back containing the measurement result value, for example `"304.2,"`.

:MEASure[:CONTinuous]:PSUPply:PCURrent

Syntax	`:MEASure[:CONTinuous]:PSUPply:PCURrent`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the peak current consumption test. To read out the latest measurement result, use the `:FETCh:PSUP:PCUR` command. Any power and current consumption measurement will need the MS Power Supply and the Current Measurement Options to be installed on your Willtek 3100.
Query	The query form of this command starts the measurement and - after the measurement has been completed and the measurement result obtained - delivers back a string containing a floating point real number representing the peak current consumption in mA.

Example	:MEAS:PSUP:PCUR? In this case, all the peak current consumption measurement is performed. As soon as the measurement has been completed, a string will be delivered back containing the measurement result value, for example "1352.9".

:MEASure:GSM:ARRay:RFTX:GROup

Syntax	:MEASure:GSM:ARRay:RFTX:GROup <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the RF TX tests, specified with the :CONF:GSM:MEAS:GRO:RFTX command for a specific number of times (set with the **int1** parameter). To read out the entire measurement results array, use the :FETCh:GSM:RFTX:GROup command. **Note:** Please keep in mind that the start of a new RF TX test will always terminate any preceding measurement of the same type instantly.
Query	The query form of this command will start the sequence of RF TX measurements as specified with the :CONF:GSM:MEAS:GRO:RFTX command for a specific number of times (set with the **int1** parameter). As soon as all measurements have been completed, a string will be delivered back, containing the related measurement result values, separated by commas. The type of the measurement result values (i.e. floating point real numbers, integers or boolean numbers) depends on the measurements specified. The order of the measurement result values within the string is as described below for the **:MEAS:GSM:CONT:RFTX:ALL** command. **Note:** Please keep in mind that the number of measurement result values delivered back is the number of measurement result values of a single execution multiplied with the **int1** parameter.
Example	**:CONF:GSM:MEAS:GRO:RFTX POW,PRMS** **:MEASure:GSM:ARRay:RFTX:GROup? 3** In this example, the group of measurements is defined by a power level measurement combined with a RMS phase error measurement. This group of measurements will be carried out three times. After all measurements have been completed, a result string will be delivered back, containing six measurement result values. The string returned is: **"4.53,9.98,4.08,10.02,4.32,9.99"**. Because of the internal order (see description of the command below), the first (third and fifth) measurement result value delivered back is the RMS phase error, while the second (fourth and sixth) measurement result value is the mobile's RF output power level.

:MEASure:GSM:ARRay:RFTX:ALL

Syntax	:MEASure:GSM:ARRay:RFTX:ALL <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.

Description

Performs a sequential measurement of the most important RF TX tests for a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFTX:ALL` command.
The RF TX tests performed by this command are:
PPEAk, the peak phase error measurement,
PRMS, the root-mean square valued phase error measurement,
FREQuency means the frequency error of the mobile under test,
LENGth the length of the burst in microseconds,
UTIMe stands for the timing error of the mobile under test in microseconds (taking the ordered timing advance into account),
POWer is the mobile's RF output power level in dBm,
TEMPlate indicates whether the power/time template has been violated by a burst or not,
CORNer gives the RF power level in dBm at eight predefined positions of a burst (see section Corner Points for details),
FLATness gives an indication whether the mobile's RF output power is constant during the active part of the burst.
Notes:

- Please keep in mind that the start of a new RF TX test will always terminate the preceding one instantly.
- For a further description of the single measurements, see description of the related commands below.

Query

The query form of this command performs the measurements and - after all measurements have been completed and all measurement results obtained - delivers a string, containing (19 * **int1**) measurement result values. The single measurement result values are separated by commas. The order and type of these measurement result values delivered back is as follows:
1. **PPEAk**, floating point real number, representing the result of the peak phase error measurement,
2. **PRMS**, floating point real number representing the result of the root-mean square valued phase error measurement,
3. **FREQuency**, floating point real number representing the mobile's frequency error,
4. **LENGth**, floating point real number representing the length of the burst in microseconds,
5. **UTIMe**, floating point real number representing the timing error of the mobile under test in microseconds (taking the ordered timing advance into account),
6. **POWer**, floating point real number representing the mobile's RF output power level in dBm,
7. **TEMPlate** boolean number. A **0** indicates that the power/time template (PTT) was **not** violated by the last burst measured, while a **1** means that the burst did violate the PTT.
8...15. **CORNer**, eight floating point real numbers, representing the RF power level in dBm at eight predefined positions of a burst.
16...19. **FLATness**, four floating point real numbers, representing the absolute minimum and maximum relative RF power levels of the mobile and their positions during the active part of the burst.

Example

```
:MEASure:GSM:ARRay:RFTX:ALL 2
:FETCh:GSM:RFTX:ALL?
```

In this case, the sequence of the most important RF TX measurements will be performed twice. All 38 measurement result values will be kept in an internal memory until they are read out with the **FETCh** command. After the internal memory has been read out, it will be cleared.
Note: A subsequent **:FETCh** command (without any previous start of a measurement) will therefore try to read nonexisting measurement data and thus create a timeout error.

:MEASure:GSM:ARRay:RFTX:PPEAk

Syntax	:MEASure:GSM:ARRay:RFTX:PPEAk <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the peak phase error for a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:GSM:RFTX:PPEAk command. For details regarding this RF TX measurement, refer to section Phase/Frequency Test Menu.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers with the physical dimension of degrees. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFTX:PPEAk? 5** The string returned is: **"5.13,5.23,4.97,5.21,4.59"**.

:MEASure:GSM:ARRay:RFTX:PRMS

Syntax	:MEASure:GSM:ARRay:RFTX:PRMS <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the root-mean square valued (RMS) phase error for a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:GSM:RFTX:PRMS command. For details regarding this RF TX measurement, refer to section Phase/Frequency Test Menu.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers with the physical dimension of degrees. The single measurement results are separated by commas.
Example	**:MEASure:ARRay:RFTX:PRMS? 5** The string returned is: **"2.33,2.53,1.97,2.41,1.79"**.

:MEASure:GSM:ARRay:RFTX:FREQuency

Syntax	:MEASure:GSM:ARRay:RFTX:FREQuency <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.

Description	Performs the measurement of the mobile's frequency error for a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFTX:FREQ` command. For details regarding this RF TX measurement, refer to section Phase/Frequency Test Menu.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers with the physical dimension of Hertz. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFTX:FREQuency 5** **:FETCh:GSM:RFTX:FREQ?**. The frequency error measurement will be performed five times. The measurement result values will be stored internally and are read out by the **FETC** command. The string returned is: **"22.33,22.53,21.97,22.41,21.79"**.

:MEASure:GSM:ARRay:RFTX:LENGth

Syntax	`:MEASure:GSM:ARRay:RFTX:LENGth <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the burst length for a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFTX:LENGth` command. For details regarding this RF TX measurement, refer to section Burst Test Menu.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers with the physical dimension of microseconds. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFTX:LENGth 5** **:FETCh:GSM:RFTX:LENG?**. The measurement of the burst length will be performed five times. The measurement result values will be stored internally and are read out by the **FETC** command. The string returned is: **"556.4,557.2,558.2,557.8,556.5"**.

:MEASure:GSM:ARRay:RFTX:UTIMe

Syntax	`:MEASure:GSM:ARRay:RFTX:UTIMe <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the uplink timing error for a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFTX:UTIMe` command. For details regarding this RF TX measurement, refer to section Burst Results Display.

Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers with the physical dimension of microseconds. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFTX:UTIMe? 5** The string returned is: **"0.0,0.1,0.0,-0.2,0.1"**.

:MEASure:GSM:ARRay:RFTX:POWer

Syntax	:MEASure:GSM:ARRay:RFTX:POWer <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **1000**. The default value for **int1** is **0**.
Description	Performs the measurement of the mobile's RF output power level a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:GSM:RFTX:POWer command. For details regarding this RF TX measurement, refer to section Burst Results Display.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers with the physical dimension of dBm. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFTX:POWer? 5** The string returned is: **"11.22,11.09,11.21,11.14,10.99"**.

:MEASure:GSM:ARRay:RFTX:TEMPlate

Syntax	:MEASure:GSM:ARRay:RFTX:TEMPlate <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs a check of the UL bursts against the power/time template (PTT) a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:GSM:RFTX:TEMPlate command. For details regarding this RF TX measurement, refer to section Burst Test Menu. The power/time template (PTT) is user-definable. For details, please refer to section Power/Time Templates (PTT). **Note:** The power/time template can also be set using the :CALC:RFTX:TEMP commands.
Query	The query form of this command will perform the check of the burst shape against the PTT the specified number of times (**int1** parameter). As soon as all checks have been completed, all check result values will be returned in a string. The string delivered back will contain **int1** boolean numbers. A **0** indicates that there was no violation of the PTT by that burst, while a **1** means that a burst did violate the PTT. The single measurement results are separated by commas.

Example	**:MEASure:GSM:ARRay:RFTX:TEMPlate 10** **:FETCh:GSM:RFTX:TEMP?**. The check of the current burst against the PTT will be performed ten times. The check result values will be stored internally and are read out by the **FETC** command. The string returned is: **"0,0,0,0,0,0,1,0,0,0"**, indicating that the seventh burst checked violated the PTT.

:MEASure:GSM:ARRay:RFTX:CORNer

Syntax	:MEASure:GSM:ARRay:RFTX:CORNer <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the mobile's RF output power level on eight user-definable positions of the ramping phases of a burst (the so-called corner points) a specific number of times (set with the **int1** parameter). To read out the entire measurement results array, use the :FETCh:GSM:RFTX:CORNer command. For details regarding this RF TX measurement, please refer to section Burst Results Display. For details on the corner points, refer to section Corner Points. **Note:** The corner points can also be set using the :CALC:RFTX:CORN commands.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain (8 * **int1**) floating point real numbers with the physical dimension of dBm. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFTX:CORNer? 2** The string returned is: "-72.18,-61.91,-20.91,-0.05,-0.04,-17.97, -56.60,-73.95,-71.38,-60.71,-20.11,-0.01, 0.04,-18.14,-63.66,-72.97".

:MEASure:GSM:ARRay:RFTX:FLATness

Syntax	:MEASure:GSM:ARRay:RFTX:FLATness <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.

Description	Performs the measurement of the flatness of the mobile's burst during the active part of it a specific number of times (set with the **int1** parameter). In short words, this measurement is a search for both the absolute minimum and the absolute maximum of the mobile's output power level during the active part of the burst (relative to the peak power). This measurement will collect four measurement result values per run: – the first value is the minimum RF power level in dB, – the second one is the position of this minimum within the burst in bit positions, – the third value is the maximum RF power level in dB, while – the last value is the position of the maximum within the burst in terms of the bit position. The resolution in the time domain is approx. 0.825 microseconds, equaling approx. 1/5 bit periods. To read out **the entire** measurement results array, use the `:FETCh:GSM:RFTX:FLATness` command.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain (4 * **int1**) floating point real numbers with the physical dimensions as explained above. The single measurement results are separated by commas.
Example	**`:MEASure:GSM:ARRay:RFTX:FLATness? 2`** The string returned is: **`"-0.12,113,7,0.56,34.0, -0.35,57.9,0.75,117.2"`**.

`:MEASure:GSM:ARRay:RFTX:FPOWer`

Syntax	`:MEASure:GSM:ARRay:RFTX:FPOWer <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **1000**. The default value for **int1** is **0**.
Description	Performs the measurement of the fast burst peak power level a specific number of times (set with the **int1** parameter). This measurement is very similar to the **:MEAS:ARR:RFTX:POW** measurement, but it is faster. To read out **the entire** measurement results array, use the `:FETCh:GSM:RFTX:FPOWer` command. For further details, refer to the description of the `:CONF:GSM:MEAS:LEV:EXP` command.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers with the physical dimension of dBm. The single measurement results are separated by commas.
Example	**`:MEASure:GSM:ARRay:RFTX:FPOWer? 5`** The string returned is: **`"13.02,13.09,13.01,13.14,13.29"`**.

:MEASure:GSM:ARRay:RFRX:RBER:ALL

Syntax `:MEASure:GSM:ARRay:RFRX:RBER:ALL <int1>`

Parameters **int1** is an integer.
The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.

Description Performs a sequential measurement of the residual bit error ratios (RBER) a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFRX:RBER:ALL` command.
The RF RX tests performed by this command are:
CIB, the residual bit error ratio of the class Ib bits,
CII, the residual bit error ratio of the class II bits, and
FER, the frame erasure ratio.
Notes:
- More details regarding RBER testing can be found in section BER Live Display.
- More details regarding bit classes can be found in section Voice Coding.
- Please keep in mind that the start of a new RF RX test will always terminate the preceding one instantly.
- The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.

Query The query form of this command will perform the measurements the specified number of times (**int1** parameter). After all measurements have been completed and all measurement results obtained, the query delivers a string, containing (3 * **int1**) measurement result values. The single measurement result values are separated by commas. All measurement result values are floating point real numbers. The order of the measurement result values delivered back is as follows:
1. **CIB**, representing the residual bit error ratio of the class Ib bits,
2. **CII**, representing the residual bit error ratio of the class II bits and
3. **FER**, representing the frame erasure ratio.

Example
```
:MEASure:GSM:ARRay:RFRX:RBER:ALL 2
:FETCh:GSM:RFRX:RBER:ALL?
```
In this case, the sequence of the residual bit error measurements will be performed twice. All six measurement result values will be kept in an internal memory until they are read out with the **FETCh** command.
The string returned is:
"0.1,1.5,0.0,0.2,2.7,0.1".

:MEASure:GSM:ARRay:RFRX:RBER:CIB

Syntax `:MEASure:GSM:ARRay:RFRX:RBER:CIB <int1>`

Parameters **int1** is an integer.
The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.

Description Performs the measurement of the residual bit error ratio of the class Ib bits a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFRX:RBER:CIB` command.
Notes:
- More details regarding RBER testing can be found in section BER Live Display.
- More details regarding bit classes can be found in section Voice Coding.
- The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.

Query	The query form of this command will perform the measurement the specified number of times (**<int1>** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **<int1>** floating point real numbers, representing the bit error ratios. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFRX:RBER:CIB? 5** The string returned is: **"0.2,0.2,0.1,0.3,0.2"**.

:MEASure:GSM:ARRay:RFRX:RBER:CII

Syntax	:MEASure:GSM:ARRay:RFRX:RBER:CII <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the residual bit error ratio of the class II bits a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:GSM:RFRX:RBER:CII command. **Notes:** – More details regarding RBER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – The number of samples to be used for testing can be set, using the :CONF:GSM:BER:COUNt command.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers, representing the bit error ratios. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFRX:RBER:CII? 5** The string returned is: **"1.4,1.4,1.5,1.7,1.3"**.

:MEASure:GSM:ARRay:RFRX:RBER:FER

Syntax	:MEASure:GSM:ARRay:RFRX:RBER:FER <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the frame erasure ratio a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:GSM:RFRX:RBER:FER command. **Notes:** – More details regarding RBER testing can be found in section BER Live Display. – The number of samples to be used for testing can be set, using the :CONF:GSM:BER:COUNt command.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers, representing the frame erasure ratios. The single measurement results are separated by commas.

Example **:MEASure:GSM:ARRay:RFRX:RBER:FER 5**
:FETCh:GSM:RFRX:RBER:FER?
The string returned is: **"0.0,0.1,0.0,0.2,0.1"**.

:MEASure:GSM:ARRay:RFRX:BER:ALL

Syntax :MEASure:GSM:ARRay:RFRX:BER:ALL <int1>

Parameters **int1** is an integer.
The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.

Description Performs a sequential measurement of the (nonresidual) residual bit error ratios (RBER) a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:GSM:RFRX:BER:ALL command.
The RF RX tests performed by this command are:
CIA, the (nonresidual) bit error ratio of the class Ia bits,
CIB, the (nonresidual) bit error ratio of the class Ib bits, and
CII, the (nonresidual) bit error ratio of the class II bits.
Notes:
- More details regarding BER testing can be found in section BER Live Display.
- More details regarding bit classes can be found in section Voice Coding.
- Please keep in mind that the start of a new RF RX test will always terminate the preceding one instantly.
- The number of samples to be used for testing can be set, using the :CONF:GSM:BER:COUNt command.

Query The query form of this command will perform the measurements the specified number of times (**int1** parameter). After all measurements have been completed and all measurement results obtained, the query delivers a string, containing (3 * **int1**) measurement result values. The single measurement result values are separated by commas. All measurement result values are floating point real numbers. The order of the measurement result values delivered back is as follows:
1. **CIA**, representing the (nonresidual) bit error ratio of the class Ia bits,
2. **CIB**, representing the (nonresidual) bit error ratio of the class Ib bits, and
3. **CII**, representing the (nonresidual) bit error ratio of the class II bits.

Example **:MEASure:GSM:ARRay:RFRX:BER:ALL 2**
:FETCh:GSM:RFRX:BER:ALL?
In this case, the sequence of the (nonresidual) bit error measurements will be performed twice. All six measurement result values will be kept in an internal memory until they are read out with the **FETCh** command.
The string returned is:
"0.1,1.5,0.0,0.2,2.7,0.1".

:MEASure:GSM:ARRay:RFRX:BER:CIA

Syntax	`:MEASure:GSM:ARRay:RFRX:BER:CIA <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the (nonresidual) bit error ratio on the class Ia bits a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFRX:BER:CIA` command. **Notes:** – More details regarding BER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers, representing the bit error ratios. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFRX:BER:CIA? 5** The string returned is: **"1.4,1.4,1.5,1.7,1.3"**.

:MEASure:GSM:ARRay:RFRX:BER:CIB

Syntax	`:MEASure:GSM:ARRay:RFRX:BER:CIB <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the (nonresidual) bit error ratio on the class Ib bits a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFRX:BER:CIB` command. **Notes:** – More details regarding BER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers, representing the bit error ratios. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFRX:BER:CIB? 5** The string returned is: **"1.4,1.4,1.5,1.7,1.3"**.

:MEASure:GSM:ARRay:RFRX:BER:CII

Syntax	`:MEASure:GSM:ARRay:RFRX:BER:CII <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the (nonresidual) bit error ratio on the class II bits a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFRX:BER:CII` command. **Notes:** – More details regarding BER testing can be found in section BER Live Display. – More details regarding bit classes can be found in section Voice Coding. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers, representing the bit error ratios. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFRX:BER:CII? 5** The string returned is: **"1.4,1.4,1.5,1.7,1.3"**.

:MEASure:GSM:ARRay:RFRX:FBER

Syntax	`:MEASure:GSM:ARRay:RFRX:FBER <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the fast bit error ratio (C-loop) test a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFRX:FBER` command. **Notes:** – More details regarding Fast BER testing can be found in section BER Live Display. – The number of samples to be used for testing can be set, using the `:CONF:GSM:BER:COUNt` command.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers, representing the bit error ratios. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFRX:FBER? 5** The string returned is: **"1.4,1.4,1.5,1.7,1.3"**.

:MEASure:GSM:ARRay:RFRX:RTDelay

Syntax	:MEASure:GSM:ARRay:RFRX:RTDelay <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the round trip delay (of data during bit error ratio measurements) a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:GSM:RFRX:RTD command. **Note:** More details regarding the round-trip delay can be found in section BER Live Display.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** integers, representing the single round trip delays. The single measurement results are separated by commas.
Example	**:MEASure:GSM:ARRay:RFRX:RTDelay? 5** The string returned is: **"8,8,8,8,8"**.

:MEAS:GSM:ARR:RFSP:ACPM:MODulation

Syntax	:MEAS:GSM:ARR:RFSP:ACPM:MODulation <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the ACPM modulation spectrum a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:GSM:RFSP:ACPM:MOD command. **Note:** According to GSM specification 05.05, this measurement will be performed on (±)100 kHz and (±)250 kHz offset from the carrier as well as on all multiples of 200 kHz offset from the carrier up to (and including) an offset of (±)1.8 MHz. If one includes the carrier frequency as well (0.0 dBc), this adds up to the 23 measurement values returned. All measurements will be performed with a measuring bandwidth of 30 kHz. More information regarding **ARRay** measurements can be found in section :MEASure:ARRay.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain (23 * **int1**) floating point real numbers. The physical dimension is dBc. The single measurement results are separated by commas. The 23 values represent the power in a 30 kHz bandwidth at the following frequencies: -1.8 MHz, -1.6 MHz, -1.4 MHz, -1.2 MHz, -1.0 MHz, -800 kHz, -600 kHz, -400 kHz, -250 kHz, -200 kHz, -100 kHz, 0 kHz, +100 kHz, 200 kHz, 250 kHz, 400 kHz,, 600 kHz, 800 kHz, 1.0 MHz, 1.2 MHz, 1.4 MHz, 1.6 MHz, 1.8 MHz.
Example	**:MEAS:GSM:ARR:RFSPectrum:ACPM:MODulation 5** **:FETCh:GSM:RFSP:ACPM:MOD?** The string returned would contain 115 floating point real numbers, representing the 23 measurement result values of five measurement runs.

:MEASure:GSM:ARRay:RFSPectrum:ACPM:MODPower

Syntax	`:MEASure[:GSM]:ARRay:RFSPectrum:ACPM:MODPower <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is `0`, the maximum value is `100`. The default value for `int1` is `0`.
Description	Performs the measurement of the ACPM modulation spectrum a specific number of times (set with the int1 parameter). To read out the entire measurement results array, use the `:FETCh:GSM:RFSP:ACPM:MODP` command. **Notes** – The first result is the absolute power in dBm, measured at the center frequency. The remaining 23 values are the same as from the `ACPM:MODulation` command. – According to GSM specification 05.05, this measurement will be performed on (±)100 kHz and (±)250 kHz offset from the carrier as well as on all multiples of 200 kHz offset from the carrier up to (and including) an offset of (±)1.8 MHz. Including the carrier frequency (0.0 dBc), this adds up to the 23 measurement values returned. All measurements will be performed with a measuring bandwidth of 30 kHz. More information regarding ARRay measurements can be found in section `:MEASure:ARRay`.
Query	The query form of this command will perform the measurement the specified number of times (`int1` parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain the absolute power at the center frequency (within a 30 kHz bandwidth) as the first value with the physical dimension of dBm and after that 23 floating point real numbers with the physical dimension dBc. The total count of the values is 24 * `int1`. The individual measurement results are separated by commas.

Example	`:MEAS:ARR:RFSP:ACPM:MODP 5`
	`:FETC:RFSP:ACPM:MODP?`
	The string returned in this example would contain 120 floating point real numbers, representing the 24 measurement result values of five measurement runs.

:MEAS:GSM:ARR:RFSP:ACPM:TRANsient

Syntax	`:MEAS:GSM:ARR:RFSP:ACPM:TRANsient <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the ACPM spectrum due to switching transients a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:GSM:RFSP:ACPM:TRAN` command. **Note:** According to GSM specification 05.05, this measurement will be performed on 0, (±)400, (±)600, (±)1.200 and (±)1.800 kHz offset from the carrier. All measurements will be performed with a measuring bandwidth of 30 kHz. For additional information, also the measurement values of the adjacent 30 kHz windows below and above the frequencies given above will be delivered back. This adds up to 9 x 3 = 27 measurement result values, that are returned per measurement run.

Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain (27 * **int1**) floating point real numbers. The physical dimension is dBm. The single measurement results are separated by commas.
Example	**:MEAS:GSM:ARR:RFSPectrum:ACPM:TRANsient 5** **:FETCh:GSM:RFSP:ACPM:TRAN?** The string returned would contain 135 floating point real numbers, representing the 27 measurement result values of five measurement runs.

:MEASure:ARRay:AFANalyser:GROup

Syntax	:MEASure:ARRay:AFANalyser:GROup <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the audio tests, specified with the :CONF:MEAS:GRO:AFAN command for a specific number of times (set with the **int1** parameter). To read out the entire measurement results array, use the :FETCh:AFAN:GROup command. **Notes:** – Please keep in mind that the start of a new AF test will always terminate all other measurements. – To perform any audio measurements on your Willtek 3100, the Audio Option must be installed.
Query	The query form of this command will start the sequence of audio measurements as specified with the :CONF:MEAS:GRO:AFAN command for a specific number of times (set with the **int1** parameter). As soon as all measurements have been completed, a string will be delivered back. It will contain floating point real numbers, representing the measurement result values. The order of the measurement result values returned is as described below (see command **:MEAS[:GSM]:ARR:AFAN:ALL**). The single measurement result values are separated by commas. **Note:** The number of measurement result values delivered back is the number of measurement result values of a single test run multiplied with the **int1** parameter.
Example	**:CONF:MEAS:GRO:AFAN SIN,FREQ** **:MEASure:ARRay:AFAN:GROup? 3** In this example, the group of measurements is defined by a SINAD measurement, combined with an AF frequency measurement. This group of measurements will be carried out three times. After all measurements have been completed, a result string will be delivered back, containing six measurement result values. The measurement result values returned are: **"1000.0,50.5,1000.1,50.1,999.9,50.6"**. Because of the internal order, the first (third and fifth) measurement result value delivered back is the audio frequency, the second (fourth and sixth) one the SINAD.

:MEASure:ARRay:AFANalyser:ALL

Syntax	:MEASure:ARRay:AFANalyser:ALL <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.

Description Performs a standard sequence of the most important audio tests a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:AFAN:ALL` command. The tests performed by this command are:
ACVPeakp, the peak-to-peak measurement of an AC voltage,
ACVRms, the RMS-valued measurement of an AC voltage,
DCVRms is used to measure AC ripple on DC lines (this measurement gives the root-mean spare voltage of the AC component of the applied DC signal).
FREQuency is the measurement of the audio frequency in Hertz,
DISTortion is the distortion measurement on the third harmonic of a sine wave and expressed in %, while
SINad is the measurement of the signal to noise ratio, expressed in dB.
Notes:
- Please keep in mind that the start of a new AF test will always terminate all other measurements.
- Any AF measurement will need the Audio Option to be installed on your Willtek 3100.

Query The query form of this command performs the measurements and - after all measurements have been completed and all measurement results obtained - delivers a string, containing (6* **int1**) measurement result values. The single measurement result values are separated by commas. All measurement result values are floating point real numbers. The order of these measurement result values delivered back is as follows:
1. **ACVPeakp**, representing the AC peak-to-peak voltage of the AF signal,
2. **ACVRms**, representing the RMS-valued AC voltage of the AF signal,
3. **DCVRms**, representing the RMS-valued AC voltage on an applied DC signal,
4. **FREQuency**, representing the audio frequency,
5. **DISTortion**, representing the third-harmonic distortion of the applied sine-wave AF signal and,
6. **SINad**, representing the signal to noise ratio of the applied AF signal.

Example **`:MEASure:ARRay:AFANalyser:ALL? 2`**
In this case, the sequence of standard AF measurements will be performed twice. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing twelve measurement result values:
`"0.7,0.25,0.0,1000.0,0.3,50.5,`
`0.75,0.3,0.0,1000.0,0.25,50.9"`.

`:MEAS:ARRay:AFAN:ACVoltage:PEAKp`

Syntax `:MEAS:ARRay:AFAN:ACVoltage:PEAKp <int1>`

Parameters **int1** is an integer.
The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.

Description Performs the measurement of the AC peak-to-peak voltage of the AF signal applied to the audio analyzer a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the `:FETCh:AFAN:ACV:PEAK` command.
Note: Any AF test command will need the Audio Option to be installed on your Willtek 3100.

Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers. The physical dimension of the measurement result values is volt, measured peak-to-peak (V_{pp}). The single measurement results are separated by commas.
Example	**:MEAS:ARR:AFANanlyser:ACVoltage:PEAKp? 5** The string returned is: **"0.7,0.8,0.75,0.71,0.79"**.

:MEAS[:GSM]:ARR:AFAN:ACVoltage:RMS

Syntax	:MEAS[:GSM]:ARR:AFAN:ACVoltage:RMS <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the RMS-valued AC voltage of the AF signal applied to the audio analyzer a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:AFAN:ACV:RMS command. **Note:** Any AF test command will need the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers. The physical dimension of the measurement result values is volt, RMS-valued (V_{rms}). The single measurement results are separated by commas.
Example	**:MEAS:GSM:ARR:AFANanlyser:ACVoltage:RMS? 5** The string returned is: **"2.52,2.55,2.42,2.49,2.51"**.

:MEAS:ARR:AFAN:DCVoltage

Syntax	:MEAS:ARR:AFAN:DCVoltage <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the RMS-valued AC component of a DC signal applied to the audio analyser a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:AFAN:DCV command. **Note:** Any AF test command will need the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers. The physical dimension of the measurement result values is volt, RMS-valued (V_{rms}). The single measurement results are separated by commas.

Example	**:MEAS:ARR:AFANanlyser:DCVoltage? 5** The string returned is: **"0.025,0.029,0.019,0.030,0.025"**.

:MEASure:ARRay:AFAN:FREQuency

Syntax	:MEASure:ARRay:AFAN:FREQuency <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the frequency of the audio signal applied to the audio analyzer a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:AFAN:FREQ command. **Note:** Any AF test command will need the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers. The physical dimension of the measurement result values is Hertz. The single measurement results are separated by commas.
Example	**:MEASure:ARRay:AFANanlyser:FREQ? 5** The string returned is: **"1000.0,1000.5,1000.9,999.9,1000.0"**.

:MEASure:ARRay:AFAN:DISTortion

Syntax	:MEASure:ARRay:AFAN:DISTortion <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the distortion of a sine wave applied to the audio analyzer a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the :FETCh:AFAN:DIST command. **Note:** Any AF test command will need the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers. The physical dimension of the measurement result values is percentage (%). The single measurement results are separated by commas.
Example	**:MEASure:ARRay:AFANanlyser:DIST? 5** The string returned is: **"0.5,0.7,1.1,1.2,0.9"**.

:MEASure:ARRay:AFANalyser:SINad

Syntax	:MEASure:ARRay:AFANalyser:SINad <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the measurement of the SINAD (i.e. signal to noise ratio) a specific number of times (set with the **int1** parameter). To read out **the entire** measurement results array, use the **:FETCh:AFAN:SIN** command. **Note:** Any AF test command will need the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command will perform the measurement the specified number of times (**int1** parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain **int1** floating point real numbers. The physical dimension of the measurement result values is dB. The single measurement results are separated by commas.
Example	**:MEASure:ARRay:AFANanlyser:SINad? 5** The string returned is: **"50.5,50.0,49.1,51.2,50.9"**.

:MEASure:ARRay:PSUPply:GROup

Syntax	:MEASure:ARRay:PSUPply:GROup <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the power supply measurements specified with the :CONF:MEAS:GRO:PSUP command for a specific number of times (set with the **int1** parameter). To read out the entire measurement results array, use the :FETCh:PSUP:GROup command. Notes – To perform any power and current consumption measurements on your Willtek 3100, the MS Power Supply and Current Measurements Option must be installed.
Query	The query form of this command will start the sequence of power/current consumption measurements as specified with the :CONF:MEAS:GRO:PSUP command for a specific number of times (set with the **int1** parameter). The order of the measurement result values returned is as described below (see command **:MEAS:PSUP:ALL**).
Example	**:CONF:MEAS:GRO:PSUP ACUR,APOW** **:MEASure:ARRay:PSUP:GROup?** In this example, the group of measurements is defined by a power consumption measurement combined with a current consumption measurement. The measurement result string returned is: **"863.6,304.2"**. Because of the internal order, the first measurement result value delivered back is the power consumption, the second one the average current consumption measurement.

:MEASure:ARRay:PSUPply:ALL

Syntax	`:MEASure:ARRay:PSUPply:ALL <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs all measurements of the power and current consumption tests for a specific number of times (set with the **int1** parameter). To read out the latest measurement results, use the `:FETCh:PSUP:ALL` command. The power supply/current measurement tests performed by this command are: **APOW** for the average power consumption measurement, measured in mW; **ACUR** for the average current consumption measurement in mA, **PCUR** for the peak current consumption in mA. **Note:** Any power and current consumption measurement will need the MS Power Supply and the Current Measurement Options to be installed on your Willtek 3100.
Query	The query form of this command starts the measurements for a specific number of times (set with the **int1** parameter). The order of these measurement result values delivered back is as follows: – **APOW**, representing the average power consumption in mW, – **ACUR**, representing the average current consumption in mA, – **PCUR**, representing the peak current consumption in mA.
Example	**:MEASure:ARRay:PSUP:ALL? 10** In this case, all power/current consumption measurements are performed ten times. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing ten times three measurement result values.

:MEASure:ARRay:PSUPply:APOWer

Syntax	`:MEASure:ARRay:PSUPply:APOWer <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the power consumption measurement for a specific number of times (set with the **int1** parameter). To read out the latest measurement results, use the `:FETCh:PSUP:APOW` command.
Query	The query form of this command starts the measurements for a specific number of times (set with the **int1** parameter). The returned string contains **int1** floating point values representing the average power consumption in mW.
Example	**:MEASure:ARRay:PSUP:APOW? 3** In this case, the power consumption measurement is performed three times. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing the three measurement result values.

:MEASure:ARRay:PSUPply:ACURrent

Syntax	`:MEASure:ARRay:PSUPply:ACURrent <int1>`

Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the average current consumption measurement for a specific number of times (set with the **int1** parameter). To read out the latest measurement results, use the `:FETCh:PSUP:ACUR` command.
Query	The query form of this command starts the measurements for a specific number of times (set with the **int1** parameter). The returned string contains **int1** floating point values representing the average current consumption in mA.
Example	**:MEASure:ARRay:PSUP:ACUR? 3** In this case, the current consumption measurement is performed three times. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing the three measurement result values.

:MEASure:ARRay:PSUPply:PCURrent

Syntax	`:MEASure:ARRay:PSUPply:PCURrent <int1>`
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	Performs the peak current consumption measurement for a specific number of times (set with the **int1** parameter). To read out the latest measurement results, use the `:FETCh:PSUP:PCUR` command.
Query	The query form of this command starts the measurements for a specific number of times (set with the **int1** parameter). The returned string contains **int1** floating point values representing the peak current consumption in mA.
Example	**:MEASure:ARRay:PSUP:PCUR? 3** In this case, the peak current consumption measurement is performed three times. As soon as all measurements have been completed and all measurement results obtained, a string will be delivered back containing the three measurement result values.

:MEASure:WCDMa[:CONTinuous]:RFTX:STOP

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:STOP`
Parameters	There are no parameters.
Description	Stops a continuous measurement of the WCDMA RF TX tests. **Note:** Stopping the test is not required since a test is automatically stopped when you start a new test.
Query	The query form of this command is not available.

Example	`:MEAS:WCDM:RFTX:FREQ`
	`:MEAS:WCDM:RFTX:STOP`
	Configures, starts and stops a measurement.

:MEASure:WCDMa[:CONTinuous]:RFTX:FREQuency

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:FREQuency`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the frequency error.
Query	The query form of this command will start the sequence of frequency error measurements.
Example	`:MEAS:WCDM:RFTX:FREQ?` Returns the measured frequency error, e.g. `-7.834`.

:MEASure:WCDMa[:CONTinuous]:RFTX:POWer:MEAN

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:POWer:MEAN`
Parameters	There are no parameters.
Description	This command starts a continuous measurement of the mobile's mean output power. To read out the latest measurement result, use the `FETCh:WCDMa:RFTX:POWer:MEAN` command.
Query	This query command will start the measurement and return a result value as soon as the first measurement has been completed. The result delivered back will contain one floating point number in dBm.
Example	`:MEA:WCDMA:RFTX:POW:MEAN?` Starts an RF power measurement and returns the result (in dBm), e.g. `-45.3`.

:MEASure:WCDMa[:CONTinuous]:RFTX:POWer:PEAK

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:POWer:PEAK`
Parameters	There are no parameters.
Description	This command starts a continuous measurement of the mobile's peak output power. To read out the latest measurement result, use the `FETCh:WCDMa:RFTX:POWer:PEAK` command.
Query	This query command will start the measurement and return a result value as soon as the first measurement has been completed. The result delivered back will contain one floating-point number in dBm.
Example	`:MEAS:WCDM:RFTX:POW:PEAK?` Starts the Peak RF power measurement and returns the result, e.g. `-36.8`.

:MEASure:WCDMa[:CONTinuous]:RFTX:NSTair[:DATa]

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:NSTair[:DATa]`
Parameters	There are no parameters.
Description	This command starts a continuous measurement of the staircase measurement. To read out the latest measurement result, use the `FETCh:WCDMa:RFTX:NSTair?` command.

Query	This query command will start the measurement and return a result value as soon as the first measurement has been completed. The result delivered back will contain 512 floating-point numbers in dBm.
Example	`:MEAS:WCDM:RFTX:NST?`

:MEASure:WCDMa[:CONTinuous]:RFTX:NSTair:POWer

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:NSTair:POWer`
Parameters	There are no parameters.
Description	This command starts a continuous measurement of the staircase power measurement. To read out the latest measurement result, use the `FETCh:WCDMa:RFTX:NSTair:POWer?` command.
Query	This query command will start the measurement and return a result value as soon as the first measurement has been completed. The result delivered will contain 7 floating point numbers in dBm.
Example	`:MEAS:WCDM:RFTX:NST:POW?`

:MEASure:WCDMa[:CONTinuous]:RFTX:ZSPan:POWer

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:ZSPan:POWer`
Parameters	There are no parameters.
Description	This command starts the zerospan analyzer measurement. To read out the latest measurement result, use the `FETCh:WCDMa:RFTX:ZSP:POW?` command.
Query	This query command will start the measurement and return a result value as soon as the measurement has been completed. The result delivered back will contain as many floating-point numbers in dBm as defined with the `RFAN:WCDM:ZSP:REFP` command.
Example	`:MEAS:WCDM:RFTX:ZSP:POW?` Starts the zerospan analyzer measurement and returns the result in a string, e.g. `-54.9,-62.7,-61.3`.

:MEASure:WCDMa[:CONTinuous]:RFTX:CPOWer

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:CPOWer`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the channel power.
Query	The query form of this command will return the float value for power.
Example	`:MEAS:WCDM:RFTX:CPOWer?` Starts a channel power measurement and returns the result (in dBm), e.g. `-45.3`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:ALL

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:ALL`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the most important RF TX Modulation Quality tests.
Query	The query form of this command starts the measurements and - after all 10 measurements have been completed and all measurement results obtained - delivers a string, containing 10 measurement result values, separated with commas. The order and type of these measurement result values delivered back is as follows: 1. `EVM RMS,` floating point real number representing the RMS vector error measurement in percent, 2. `EVM Peak`, floating point real number representing the peak vector error measurement in percent, 3. `Magnitude error RMS`, floating point real number representing the RMS magnitude vector error measurement in percent, 4. `Magnitude error Peak`, floating point real number representing the PEAK magnitude vector error measurement in percent, 5. `Phase error RMS`, floating point real number representing the RMS phase vector error measurement in degree, 6. `Phase error Peak`, floating point real number representing the PEAK phase vector error measurement in degree 7. `Frequency error`, floating point real number representing the mobile's frequency error, 8. `RHO`, floating point real number representing the mobile's modulation quality, 9. `I/Q Offset`, a floating point value with the result in dBc, representing the result of the origin offset vector error measurement, 10. `I/Q Imbalance`, floating point real number representing the result of the IQImbalance vector error measurement with the result in dB. **Note:** For a further description of the single measurements, see description of the related commands below.
Example	`:MEAS:WCDM:RFTX:MODQ:ALL?` In this case, all relevant RFTX measurements will be performed in a sequence, e.g. `21.624,72.8,15.335,72.736,6.738,27.512,-6.378,0.961,-28.251,-43.743`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:ERMS

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:ERMS`
Parameters	There are no parameters.
Description	Starts a continuous RMS vector error magnitude measurement.
Query	The query form of this command will return a floating point value; the unit is percent.
Example	`:MEAS:WCDM:RFTX:MODQ:ERMS?` Returns the measured EVM RMS vector error, e.g. `21.624`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:EPEAk

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:EPEAk`
Parameters	There are no parameters.
Description	Starts a continuous peak vector error magnitude measurement.
Query	The query form of this command will return a floating point value in percent.
Example	`:MEAS:WCDM:RFTX:MODQ:EPEA?` Returns the measured EVM peak vector error, e.g. `72.8`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:MRMS

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:MRMS`
Parameters	There are no parameters.
Description	Starts a continuous RMS magnitude measurement.
Query	The query form of this command will return a floating point value; the unit is percent.
Example	`:MEAS:WCDM:RFTX:MODQ:MRMS?` Returns the measured Magnitude RMS vector error, e.g. `15.335`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:MPEAk

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:MPEAk`
Parameters	There are no parameters.
Description	Starts a continuous peak magnitude measurement.
Query	The query form of this command will return a floating point value in percent.
Example	`:MEAS:WCDM:RFTX:MODQ:MPEA?` Returns the measured Magnitude peak vector error, e.g. `72.736`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:PRMS

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:PRMS`
Parameters	There are no parameters.
Description	Starts a continuous RMS phase error measurement.
Query	The query form of this command will return a floating point value; the unit is percent.
Example	`:MEAS:WCDM:RFTX:MODQ:PRMS?` Returns the measured RMS phase vector error, e.g. `6.738`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:PPEAk

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:PPEAk`
Parameters	There are no parameters.
Description	Starts a continuous peak phase error measurement.
Query	The query form of this command will return a floating point value in percent.
Example	`:MEAS:WCDM:RFTX:MODQ:PPEA?` Returns the measured phase peak vector error, e.g. `27.512`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:RHO

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:RHO`
Parameters	There are no parameters.
Description	Starts a continuous closed loop measurement.
Query	The query form of this command will return a floating point value.
Example	`:MEAS:WCDM:RFTX:MODQ:RHO?` Returns the modulation quality, e.g. `0.9989`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality :IQOFfset

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:IQOFfset`
Parameters	There are no parameters.
Description	Starts a continuous IQ origin offset vector error measurement.
Query	The query form of this command will return a floating point value with the result in dBc.
Example	`:MEAS:WCDM:RFTX:MODQ:IQOF?` Returns the origin offset in dBc, e.g. `-28.251`.

:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality :IQIMbalance

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFTX:MODQuality:IQIMbalance`
Parameters	There are no parameters.
Description	Starts a continuous IQ imbalance vector error measurement.
Query	The query form of this command will return a floating point value with the result in dB.
Example	`:MEAS:WCDM:RFTX:MODQ:IQIM?` Returns the IQ imbalance in dBc, e.g. `-43.743`.

:MEASure:WCDMa[:CONTinuous]:RFSPectrum:MSPectrum[:DATa]

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFSPectrum:MSPectrum[:DATa]`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the RF modulation spectrum. To read out the latest measurement result, use the `FETC:WCDM:RFSP:MSP[:DATA]` command.
Query	This query command will start the measurement and return a result value as soon as the first measurement has been completed. The result delivered will contain an array of 501 floating point numbers.
Example	`:MEAS:WCDM:RFSP:MSP?`

:MEASure:WCDMa[:CONTinuous]:RFSPectrum:MSPectrum:AVG

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFSPectrum:MSPectrum:AVG <Int1>`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the RF modulation spectrum averaged over `<Int1>` measurements. To read out the latest measurement result use the `FETC:WCDM:RFSP:MSP:AVG` command.
Query	This query command will start the measurement and return a result value as soon as `<Int1>` measurements have been completed. The result delivered will contain an array of 501 floating point numbers.
Example	`:MEAS:WCDM:RFSP:MSP:AVG 10?`

:MEASure:WCDMa[:CONTinuous]:RFSPectrum:MSPectrum:OBW

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFSPectrum:MSPectrum:OBW`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the occupied bandwidth. To read out the latest measurement result use the `FETC:WCDM:RFSP:MSP:OBW` command.
Query	This query command will start the measurement and return a result value as soon as the first measurement have been completed. The result delivered will contain one floating point number.
Example	`:MEAS:WCDM:RFSP:OBW?`

:MEAS:WCDMA[:CONTinuous]:RFSPectrum:ACLR

Syntax	`:MEAS:WCDMA[:CONTinuous]:RFSPectrum:ACLR`
Parameters	There are no parameters.
Description	This command starts the adjacent channel leakage power ratio measurement. To read out the latest measurement result use the `FETC:WCDM:RFSP:ACLR?` command.
Query	The query form of this command starts the measurement and delivers a string, containing 5 measurement result values separated by commas (12 offsets in both directions from the carrier, plus the result at the carrier itself). These measurement result values are floating point real numbers with the physical dimension dBc (dB relative to the carrier, being the transmission frequency of the mobile).
Example	`:MEAS:WCDM:RFSP:ACLR?`

:MEASure:WCDMa[:CONTinuous]:RFSPectrum:SEM[:DATa]

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFSPectrum:SEM[:DATa]`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the Spectrum Emission Mask. To read out the latest measurement result use the `FETC:WCDM:RFSP:SEM[:DATA]` command.
Query	This query command will start the measurement and return a result value as soon as the first measurement has been completed. The result delivered back will contain an array of 501 floating point numbers.
Example	`:MEAS:WCDM:RFSP:SEM?`

:MEASure:WCDMa[:CONTinuous]:RFSPectrum:SEM:AVG

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFSPectrum:SEM:AVG <Int1>`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the RF modulation spectrum averaged over `<Int1>` measurements. To read out the latest measurement result use the `FETC:WCDM:RFSP:SEM:AVG` command.
Query	This query command will start the measurement and return a result value as soon as `<Int1>` measurements have been completed. The result delivered back will contain an array of 501 floating point numbers.
Example	`:MEAS:WCDM:RFSP:SEM:AVG 10?`

:MEASure:WCDMa[:CONTinuous]:RFRX:STOP

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFRX:STOP`
Parameters	There are no parameters.
Description	Stops a continuous measurement of the WCDMA RF RX tests. **Note:** Stopping the test is not required since a test is automatically stopped when you start a new test.
Query	The query form of this command is not available.

Example	`:MEAS:WCDMa:RFRX:FREQ`
	`:MEAS:WCDMa:RFRX:STOP`
	Configures, starts and stops a measurement.

:MEASure:WCDMa[:CONTinuous]:RFRX:STOP

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFRX:STOP`
Parameters	There are no parameters.
Description	Stops a continuous measurement of the WCDMA RF RX tests. **Note:** Stopping the test is not required since a test is automatically stopped when you start a new test.
Query	The query form of this command is not available.

Example	`:MEAS:WCDMa:RFRX:FREQ`
	`:MEAS:WCDMa:RFRX:STOP`
	Configures, starts and stops a measurement.

:MEASure:WCDMa[:CONTinuous]:RFRX:BER:RATIo

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFRX:BER:RATIo`
Parameters	There are no parameters.
Description	Starts a continuous BER measurement.
Query	The query form of this command will return a string containing the floating point value with the BER result.
Example	`:MEAS:WCDM:RFRX:BER?` Starts the measurement and returns the result, e.g. `3.1`.

:MEASure:WCDMa[:CONTinuous]:RFRX:BLER:RATIo

Syntax	`:MEASure:WCDMa[:CONTinuous]:RFRX:BLER:RATIo`
Parameters	There are no parameters.
Description	Starts a continuous BLER measurement.
Query	The query form of this command will return a string containing the floating point value with the BLER result.

Example	:MEAS:WCDM:RFRX:BLER? Starts the measurement and returns the result, e.g. 3.1.

:MEASure:WCDMa[:CONTinuous]:UEReport:STOP

Syntax	:MEASure:WCDMa[:CONTinuous]:UEReport:STOP
Parameters	There are no parameters.
Description	Stops a continuous measurement of the WCDMA UE Report. **Note:** Stopping the test is not required since a test is automatically stopped when you start a new test.
Query	The query form of this command is not available.
Example	:MEAS:WCDM:UER:PLOS :MEAS:WCDM:UER:STOP Configures, starts and stops a measurement.

:MEASure:WCDMa[:CONTinuous]:UEReport:ALL

Syntax	:MEASure:WCDMa[:CONTinuous]:UEReport:ALL
Parameters	There are no parameters.
Description	Starts a continuous measurement of all available UE Report tests.
Query	The query form of this command starts the measurements and - after all 6 measurements have been completed and all measurement results obtained - delivers a string, containing 6 measurement result values, separated with commas. The order and type of these measurement result values delivered back is as follows: 1. CPICh ECNO, floating point real number representing UE measured normalized chip energy per noise value of the P-CPICH, 2. CPICh RSCP, floating point real number representing UE measured received signal code power of the P-CPICH, 3. PLOSs, floating point real number representing the pathloss between Node-B and UE, estimated by the UE, 4. TXPower, floating point real number representing the UE transmit power, 5. RTTDifference, floating point real number representing UE measured RX-Time-Difference between UL and DL signal, 6. TRCH BLER, floating point real number representing the UE estimated BLER value for the Transport Channel defined with CONFigure:WCDMa:MEASure:UEReport:TRCH. **Note:** For a further description of the single measurements, see description of the related commands below.
Example	:MEAS:WCDM:UER:ALL? All UE Report measurements will be performed in a sequence, e.g. 23,85,102,97,992,3.

:MEASure:WCDMa[:CONTinuous]:UEReport:CPICh:ECNO

Syntax	`:MEASure:WCDMa[:CONTinuous]:UEReport:CPICh:ECNO`
Parameters	There are no parameters.
Description	Reports the UE measured normalized chip energy per noise value of the P-CPICH.
Query	The query form of this command will return a string containing the floating point value with the result.
Example	`:MEAS:WCDM:UER:CPICh:ECNO?` Starts the measurement and returns the result, e.g. `31`.

:MEASure:WCDMa[:CONTinuous]:UEReport:CPICh:RSCP

Syntax	`:MEASure:WCDMa[:CONTinuous]:UEReport:CPICh:RSCP`
Parameters	There are no parameters.
Description	Reports the UE measured received signal code power of the P-CPICH.
Query	The query form of this command will return a string containing the floating point value with the result.
Example	`:MEAS:WCDM:UER:CPICh:RSCP?` Starts the measurement and returns the result, e.g. `41`.

:MEASure:WCDMa[:CONTinuous]:UEReport:PLOSs

Syntax	`:MEASure:WCDMa[:CONTinuous]:UEReport:PLOSs`
Parameters	There are no parameters.
Description	Reports the path loss between Node B and UE, estimated by the UE.
Query	The query form of this command will return a string containing the floating point value with the result.
Example	`:MEAS:WCDM:UER:PLOS?` Starts the measurement and returns the result, e.g. `61`.

:MEASure:WCDMa[:CONTinuous]:UEReport:TXPOwer

Syntax	`:MEASure:WCDMa[:CONTinuous]:UEReport:TXPOwer`
Parameters	There are no parameters.
Description	Reports the UE transmit power.
Query	The query form of this command will return a string containing the floating point value with the result.
Example	`:MEAS:WCDM:UER:TXPO?` Starts the measurement and returns the result, e.g. `66`.

:MEASure:WCDMa[:CONTinuous]:UEReport:RTTDifference

Syntax	`:MEASure:WCDMa[:CONTinuous]:UEReport:RTTDifference`
Parameters	There are no parameters.
Description	Reports the UE measured RX-Time-Difference between UL and DL signal.
Query	The query form of this command will return a string containing the floating point value with the result.
Example	`:MEAS:WCDM:UER:RTTD?` Starts the measurement and returns the result, e.g. `1056`.

:MEASure:WCDMa[:CONTinuous]:UEReport:TRCH:BLER

Syntax	`:MEASure:WCDMa[:CONTinuous]:UEReport:TRCH:BLER`
Parameters	There are no parameters.
Description	Reports the UE estimated BLER value for the Transport Channel defined with `CONFigure:WCDMa:MEASure:UEReport:TRCH`.
Query	The query form of this command will return a string containing the floating point value with the result.
Example	`:MEAS:WCDM:UER:TRCH:BLER?` Starts the measurement and returns the result, e.g. `5`.

:MEASure:WCDMa:ARRay:RFTX:FREQuency

Syntax	`:MEASure:WCDMa:ARRay:RFTX:FREQuency <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs the frequency error measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times and return the results.
Example	`:MEAS:WCDM:ARR:RFTX:FREQ? 5` Starts five measurements, the results of which are returned. Example: `47.1,91.5,13.6,-4.8,15.3`.

:MEASure:WCDMa:ARRay:RFTX:POWer:MEAN

Syntax	`:MEASure:WCDMa:ARRay:RFTX:POWer:MEAN <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	This command starts a continuous array measurement of the mobile's output power specified by number of arrays `<Int1>`. To read out the entire measurement results array use the command `FETCh:WCDMa:RFTX:POWer:MEAN`.

Query	This query command will perform the measurement the number of times specified by <Int1>. As soon as all measurements have been completed, all result values will be returned. The values delivered will contain Int1 floating-point numbers in dBm.
Example	:MEAS:WCDM:ARR:RFTX:POW:MEAN? 10 Starts ten measurements, the results of which are returned. Example: 47.1,91.5,13.6,-4.8,15.3,37.1,51.5,33.6,-4.4,55.3

:MEASure:WCDMa:ARRay:RFTX:POWer:PEAK

Syntax	:MEASure:WCDMa:ARRay:RFTX:POWer:PEAK <Int1>
Parameters	Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0.
Description	This command starts a continuous array measurement of the mobile's peak output power specified by number of arrays <Int1>. To read out the entire measurement results array use the command FETCh:WCDMa:RFTX:POWer:PEAK.
Query	This query command will perform the measurement the specified number of <Int1>. As soon as all measurements have been completed, all result values will be returned. The values delivered back will contain <Int1> floating-point numbers in dBm.
Example	:MEAS:WCDM:ARR:RFTX:POW:PEAK? 10

:MEASure:WCDMa:ARRay:RFTX:NSTair[:DATa]

Syntax	:MEASure:WCDMa:ARRay:RFTX:NSTair[:DATa]
Parameters	There are no parameters.
Description	This command starts a continuous measurement of the Staircase measurement. To read out the latest measurement result, use the FETCh:WCDMa:RFTX:NSTair? command.
Query	This query command will start the measurement and return a result value as soon as the first measurement has been completed. The result delivered back will contain 512 floating-point numbers in dBm.
Example	:MEAS:WCDM:ARRay:RFTX:NST?

:MEASure:WCDMa:ARRay:RFTX:NSTair:POWer

Syntax	:MEASure:WCDMa:ARRay:RFTX:NSTair:POWer
Parameters	There are no parameters.
Description	This command starts a continuous measurement of the staircase power measurement. To read out the latest measurement result use the FETCh:WCDMa:RFTX:NSTair:POWer? command.
Query	This query command will start the measurement and return a result value as soon as the first measurement has been completed. The result delivered back will contain 7 floating point numbers in dBm.
Example	:MEAS:WCDM:ARRay:RFTX:NST:POW?

:MEASure:WCDMa:ARRay:RFTX:CPOWer

Syntax	`:MEASure:WCDMa:ARRay:RFTX:CPOWer <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs the measurement of the channel power as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the result (in dBm).
Example	`:MEAS:WCDM:ARR:RFTX:CPOW? 10`

:MEASure:WCDMa:ARRay:RFTX:MODQuality:ALL

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:ALL <Int1>`
Parameters	There are no parameters.
Description	Starts an array measurement of the most important RF TX Modulation Quality tests as many times as specified.
Query	The query form of this command starts the measurements and - after all 10 measurements have been completed and all measurement results obtained - delivers a string, containing 10 measurement result values, separated with commas. The order and type of these measurement result values delivered is as follows: 1. `EVM RMS`, floating point real number representing the RMS vector error measurement in percent, 2. `EVM Peak`, floating point real number representing the peak vector error measurement in percent, 3. `Magnitude error RMS`, floating point real number representing the RMS magnitude vector error measurement in percent, 4. `Magnitude error Peak`, floating point real number representing the PEAK magnitude vector error measurement in percent, 5. `Phase error RMS`, floating point real number representing the RMS phase vector error measurement in degree, 6. `Phase error Peak`, floating point real number representing the PEAK phase vector error measurement in degree, 7. `Frequency error`, floating point real number representing the mobile's frequency error, 8. `RHO`, floating point real number representing the mobile's modulation quality, 9. `I/Q Offset`, a floating point value with the result in dBc, representing the result of the origin offset vector error measurement, 10. `I/Q Imbalance`, floating point real number representing the result of the IQ Imbalance vector error measurement with the result in dB. **Note:** For a further description of the single measurements, see description of the related commands below.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:ALL? 5` In this case, all relevant RF TX measurements will be performed in a sequence.

:MEASure:WCDMa:ARRay:RFTX:MODQuality:ERMS

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:ERMS <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs an RMS vector error measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:ERMS? 19`

:MEASure:WCDMa:ARRay:RFTX:MODQuality:EPEAk

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:EPEAk <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs a peak vector error measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:EPEA? 10`

:MEASure:WCDMa:ARRay:RFTX:MODQuality:MRMS

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:MRMS <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs an RMS magnitude vector error measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:MRMS? 10`

:MEASure:WCDMa:ARRay:RFTX:MODQuality:MPEAk

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:MPEAk <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs a peak magnitude vector error measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:MPEA? 10`

:MEASure:WCDMa:ARRay:RFTX:MODQuality:PHASe:RMS

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:PHASe:RMS <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs an RMS phase vector error measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:PRMS?`

:MEASure:WCDMa:ARRay:RFTX:MODQuality:PPEAk

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:PPEAk <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs a peak phase vector error measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:PPEA? 10`

:MEASure:WCDMa:ARRay:RFTX:MODQuality:RHO

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:RHO <Int1>`
Parameters	`Int1` is an integer. The minimum value is `0`, the maximum value is `100`. The default value is `0`.
Description	Starts a continuous closed loop measurement.
Query	The query form of this command will return a floating point value.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:RHO? 5`

:MEASure:WCDMa:ARRay:RFTX:MODQuality:IQOFfset

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:IQOFfset <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs an origin offset vector error measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:IQOF? 10`

:MEASure:WCDMa:ARRay:RFTX:MODQuality:IQIMbalance

Syntax	`:MEASure:WCDMa:ARRay:RFTX:MODQuality:IQIMbalance <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs an IQ imbalance vector error measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFTX:MODQ:IQIM? 10`

:MEASure:WCDMa:ARRay:RFSPectrum:MSPectrum:OBW

Syntax	`:MEASure:WCDMa:ARRay:RFSPectrum:MSPectrum:OBW`
Parameters	There are no parameters.
Description	Performs an occupied bandwidth measurement as many times as specified. To read out the latest measurement result use the `FETC:WCDM:RFSP:MSP:OBW` command.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFSP:OBW? 15`

:MEASure:WCDMa:ARRay:RFSPectrum:ACLR

Syntax	`:MEASure:WCDMa:ARRay:RFSPectrum:ACLR <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is `0`, the maximum value is `100`. The default value for `int1` is `0`.
Description	Performs the measurement of the ACLR modulation spectrum a specific number of times (set with the `int1` parameter).
Query	The query form of this command will perform the measurement the specified number of times (`int1` parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered will contain (25 * `int1`) floating point real numbers. The physical dimension is dBc. The single measurement results are separated by commas.

Example	`:MEAS:WCDM:ARR:RFSP:ACLR 10`
	`:FETCh:WCDMA:RFSP:ACLR`
	The string returned in this example would contain 50 floating point real numbers, representing the 5 measurement result values of 10 measurement runs.

:MEASure:WCDMa:ARRay:RFRX:BER:RATIo

Syntax	`:MEASure:WCDMa:ARRay:RFRX:BER:RATIo <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs a BER measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFRX:BER? 100`

:MEASure:WCDMa:ARRay:RFRX:BLER:RATIo

Syntax	`:MEASure:WCDMa:ARRay:RFRX:BLER:RATIo <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Performs a BLER measurement as many times as specified.
Query	The query form of this command will perform the measurement the specified number of times, and return the results.
Example	`:MEAS:WCDM:ARR:RFRX:BLER? 100`

:MEASure:WCDMa:ARRay:UEReport:ALL

Syntax	`:MEASure:WCDMa:ARRay:UEReport:ALL`
Parameters	There are no parameters.
Description	Starts an array measurement of all available UE Report tests.
Query	The query form of this command starts the measurements and - after all 6 measurements have been completed and all measurement results obtained - delivers a string, containing 6 measurement result values, separated with commas. The order and type of these measurement result values delivered is as follows: 1. `CPICh ECNO`, floating point real number representing UE measured normalized chip energy per noise value of the P-CPICH, 2. `CPICh RSCP`, floating point real number representing UE measured received signal code power of the P-CPICH, 3. `PLOSs`, floating point real number representing the path loss between Node B and UE, estimated by the UE, 4. `TXPower`, floating point real number representing the UE transmit power, 5. `RTTDifference`, floating point real number representing UE measured RX-Time-Difference between UL and DL signal, 6. `TRCH BLER`, floating point real number representing the UE estimated BLER value for the Transport Channel defined with `CONFigure:WCDMa:MEASure:UEReport:TRCH`. **Note:** For a further description of the single measurements, see description of the related commands below.

Example	:MEAS:WCDM:ARR:UER:ALL? 2 All UE Report measurements will be performed in a sequence, e.g. 23,85,102,97,992,3,23,85,102,97,992,3.

:MEASure:WCDMa:ARRay:UEReport:CPICh:ECNO

Syntax	:MEASure:WCDMa:ARRay:UEReport:CPICh:ECNO <Int1>
Parameters	Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0.
Description	Reports the UE measured normalized chip energy per noise value of the P-CPICH.
Query	The query form of this command will perform the measurement the specified number of times and return the results.
Example	:MEAS:WCDM:ARR:UER:CPICh:ECNO? 7 Starts the measurement and returns the result, e.g. 31,43,16,5,43,28,31.

:MEASure:WCDMa:ARRay:UEReport:CPICh:RSCP

Syntax	:MEASure:WCDMa:ARRay:UEReport:CPICh:RSCP <Int1>
Parameters	Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0.
Description	Reports the UE measured received signal code power of the P-CPICH.
Query	The query form of this command will perform the measurement the specified number of times and return the results.
Example	:MEAS:WCDM:ARR:UER:CPICh:RSCP? 6 Starts the measurement and returns the result, e.g. 31,43,66,75,43,28.

:MEASure:WCDMa:ARRay:UEReport:PLOSs

Syntax	:MEASure:WCDMa:ARRay:UEReport:PLOSs <Int1>
Parameters	Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0.
Description	Reports the path loss between Node B and UE, estimated by the UE.
Query	The query form of this command will perform the measurement the specified number of times and return the results.
Example	:MEAS:WCDM:ARR:UER:PLOS? 3 Starts the measurement and returns the result, e.g. 75,46,128.

:MEASure:WCDMa:ARRay:UEReport:TXPOwer

Syntax	`:MEASure:WCDMa:ARRay:UEReport:TXPOwer <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Reports the UE transmit power.
Query	The query form of this command will perform the measurement the specified number of times and return the results.
Example	`:MEAS:WCDM:ARR:UER:TXPO? 4` Starts the measurement and returns the result, e.g. `66,75,43,28`.

:MEASure:WCDMa:ARRay:UEReport:RTTDifference

Syntax	`:MEASure:WCDMa:ARRay:UEReport:RTTDifference <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Reports the UE measured RX-Time-Difference between UL and DL signal.
Query	The query form of this command will perform the measurement the specified number of times and return the results.
Example	`:MEAS:WCDM:ARR:UER:RTTD? 5` Starts the measurement and returns the result, e.g. `854,966,1075,943,1128`.

:MEASure:WCDMa:ARRay:UEReport:TRCH:BLER

Syntax	`:MEASure:WCDMa:ARRay:UEReport:TRCH:BLER <Int1>`
Parameters	`Int1` is an integer. The minimum value for `Int1` is `0`, the maximum value is `100`. The default value is `0`.
Description	Reports the UE estimated BLER value for the Transport Channel defined with `CONFigure:WCDMa:MEASure:UEReport:TRCH`.
Query	The query form of this command will perform the measurement the specified number of times and return the results.
Example	`:MEAS:WCDM:ARR:UER:TRCH:BLER? 2` Starts the measurement and returns the result, e.g. `0,2`.

FETCh Subsystem

The commands of this subsystem are used to read out measurement result values both from continuous and array measurements.

:FETCh:LAST?

Syntax	:FETCh:LAST?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns in a string the latest measurement result value(s) of the measurement started last. The number and format of the measurement result values returned is dependent on the type of measurement.
Example	**:MEAS:GSM:RFTX:PRMS** **:MEAS:GSM:RFRX:RBER:ALL** **:FETCh:LAST?** In this example, the **FETC:LAST** command will return the three residual bit error ratios (three floating point real numbers) as this is the measurement started last.

:FETCh:GSM:RFTX:GROup?

Syntax	:FETCh:GSM:RFTX:GROup?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result values of the sequence of RF TX measurements as specified with the :CONF:MEAS:GRO:RFTX command and measured with the :MEAS:GSM[:ARRay]:RFTX:GROup command. The string returned contains the related measurement result values. The type of the measurement result values (i.e. floating point real numbers, integers or boolean numbers) depends on the kind of measurements. The order of the measurement result values within the string is as described below for the **:FETCh:GSM:RFTX:ALL** command.
Example	**:CONF:MEAS:GRO:RFTX POW,PRMS** **:MEAS:GSM:RFTX:GRO** **:FETCh:GSM:RFTX:GROup?** In this example, the group of measurements is defined by a power measurement combined with a RMS phase error measurement. The values returned in this example are: **"4.53,9.98"**. Because of the internal order (see description of the command below), the first measurement result value delivered back is the RMS phase error, the second one the mobile's RF output power.

:FETCh:GSM:RFTX:ALL?

Syntax	:FETCh:GSM:RFTX:ALL?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result values of the :MEAS:GSM[:ARRay]:RFTX:ALL measurement. The string delivered back will contain 19 measurement result values, separated by commas. The order and type of these measurement result values delivered back is as follows: 1. **PPEAk**, floating point real number, representing the peak phase error measurement, 2. **PRMS**, floating point real number representing the root-mean square value of the phase error, 3. **FREQuency**, floating point real number representing the mobile's frequency error, 4. **LENGth**, floating point real number representing the length of the burst in microseconds, 5. **UTIMe**, floating point real number representing the timing error of the mobile under test in microseconds (taking the ordered timing advance into account), 6. **POWer**, floating point real number representing the mobile's RF output power level in dBm, 7. **TEMPlate** boolean number. A **0** indicates that the power/time template was **not** violated by a burst while a **1** means that there was a violation of the power/time template. 8...15. **CORNer**, eight floating point real numbers, representing the RF power level in dBm at eight predefined positions of a burst (see section Corner Points for details). 16...19. **FLATness**, four floating point real numbers, representing the absolute minimum and maximum RF power levels of the mobile and their positions during the active part of the burst.
Example	**:MEAS:GSM:RFTX:ALL** **:FETCh:GSM:RFTX:ALL?** This example will return the latest measurement result values for the 19 most important RF TX measurements. The string delivered back is: **"5.13,1.94,-2.22,557.0,0.1,11.22,0,-72.18, -61.91,-20.91,-0.05,-0.04,-17.97,-56.60, -73.95,-0.05,-0.01,0.02,-0.05"**.

:FETCh:GSM:RFTX:PPEAk?

Syntax	:FETCh:GSM:RFTX:PPEAk?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result value(s) of the MEAS:GSM[:ARRay]:RFTX:PPEAk measurement. The string delivered back will contain at least one floating point real number with the physical dimension of degrees.
Example	**:MEAS:GSM:RFTX:PPEAk** **:FETCh:GSM:RFTX:PPEAk?** The value returned is: **"5.13"**.

:FETCh:GSM:RFTX:PRMS?

Syntax	:FETCh:GSM:RFTX:PRMS?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result value(s) of the MEAS:GSM[:ARRay]:RFTX:PRMS measurement. The string delivered back will contain at least one floating point real number with the physical dimension of degrees.
Example	**:MEAS:GSM:ARR:RFTX:PRMS 5** **:FETCh:GSM:RFTX:PRMS?** The values returned are: **"2.49,2.37,2.51,2.47,2.35"**.

:FETCh:GSM:RFTX:FREQuency?

Syntax	:FETCh:GSM:RFTX:FREQuency?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result value(s) of the MEAS:GSM[:ARRay]:RFTX:FREQ measurement. The string delivered back will contain at least one floating point real number with the physical dimension of Hertz.
Example	**:MEAS:GSM:ARR:RFTX:FREQ 5** **:FETCh:GSM:RFTX:FREQ?** The values returned are: **"22.49,24.37,21.51,29.47,23.35"**.

:FETCh:GSM:RFTX:LENGth?

Syntax	:FETCh:GSM:RFTX:LENGth?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result value(s) of the MEAS:GSM[:ARRay]:RFTX:LENGth measurement. The string delivered back will contain at least one floating point real number with the physical dimension of microseconds.
Example	**:MEAS:GSM:RFTX:LENGth** **:FETCh:GSM:RFTX:LENG?** The value returned is: **"557.0"**.

:FETCh:GSM:RFTX:UTIMe?

Syntax	:FETCh:GSM:RFTX:UTIMe?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result value(s) of the MEAS:GSM[:ARRay]:RFTX:UTIMe measurement. The string delivered back will contain at least one floating point real number with the physical dimension of microseconds.
Example	**:MEAS:GSM:ARR:RFTX:UTIMe 5** **:FETCh:GSM:RFTX:UTIM?** The values returned are: **"1.1,0.8,1.2,0.5,0.2"**.

:FETCh:GSM:RFTX:POWer?

Syntax	:FETCh:GSM:RFTX:POWer?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result value(s) of the MEAS:GSM[:ARRay]:RFTX:POWer measurement. The string delivered back will contain at least one floating point real number with the physical dimension of dBm.
Example	**:MEAS:GSM:RFTX:POWer** **:FETCh:GSM:RFTX:POWer?** The value returned is: **"13.05"**.

:FETCh:GSM:RFTX:TEMPlate?

Syntax	:FETCh:GSM:RFTX:TEMPlate?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest result value(s) of the :MEAS:GSM[:ARRay]:RFTX:TEMPlate check. The string delivered back will contain at least one boolean number, representing the result of the check. A **0** indicates that the burst measured was within the PTT specified, while a **1** means that the burst did violate the PTT. The PTT is user-definable and can be set using the :CALC:GSM:RFTX:TEMP commands.
Example	**:MEAS:GSM:ARR:RFTX:TEMPlate 10** **:FETCh:GSM:RFTX:TEMP?** The values returned are: **"0,0,0,0,0,0,0,0,0,0"**. This means that all the ten bursts checked were within the PTT.

:FETCh:GSM:RFTX:CORNer?

Syntax	:FETCh:GSM:RFTX:CORNer?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result values of the :MEAS:GSM[:ARRay]:RFTX:CORNer measurement. The string delivered back will contain at least eight floating point real numbers with the physical dimension of dBm.
Example	**:MEAS:GSM:RFTX:CORNer** **:FETCh:GSM:RFTX:CORN?** The values returned are: **"-72.18,-61.91,-20.91,-0.05, -0.04,-17.97,-56.60,-73.95"**.

:FETCh:GSM:RFTX:FLATness?

Syntax	:FETCh:GSM:RFTX:FLATness?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result values of the :MEAS:GSM[:ARRay]:RFTX:FLATness measurement. The string delivered back will contain at least four floating point real numbers with the physical dimensions of dBm (first and third number) and bit periods (second and fourth number).
Example	**:MEAS:GSM:RFTX:FLATness** **:FETCh:GSM:RFTX:FLAT?** The values returned are: **"-0.12,113.7,0.56,34.0"**.

:FETCh:GSM:RFTX:FPOWer?

Syntax	:FETCh:GSM:RFTX:FPOWer?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result value(s) of the :MEAS:GSM[:ARRay]:RFTX:FPOWer measurement. The string delivered back will contain at least one floating point real number with the physical dimension of dBm.
Example	**:MEAS:GSM:RFTX:FPOWer** **:FETCh:GSM:RFTX:FPOWer?** The value returned is: **"13.05"**.

:FETCh[:GSM]:RFTX:LPOWer?

Syntax	:FETCh[:GSM]:RFTX:LPOWer?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest measurement result value(s) of the :MEAS:GSM:ARRay:RFTX:LPOWer measurement. The string delivered back will contain at least one floating point real number with the physical dimension of dBm.
Example	:MEAS:RFTX:LPOW :FETC:RFTX:LPOW? The value returned in this example is: "13.05".

:FETCh:GSM:BLOCkdata:BURStshape?

Syntax	:FETCh:GSM:BLOCkdata:BURStshape?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest data block describing the burst shape. For further details regarding this measurement, please refer to the description of the :MEAS:GSM:BLOC:BURS command.
Example	**:MEAS:GSM:BLOC:BURS** **:FETC:GSM:BLOC:BURS?** The **FETC** command will return a data array, containing 711 floating point real numbers.

:FETCh:GSM:BLOCkdata:PHASeerror?

Syntax	:FETCh:GSM:BLOCkdata:PHASeerror?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest data block describing the graph of the phase error. For further details regarding this measurement, please refer to the description of the :MEAS:GSM:BLOC:PHAS command and the explanations given in section :MEASure:BLOCkdata. There, you will also find all information about the number, type and order of the measurement result values returned in the result string.
Example	**:MEAS:GSM:BLOC:PHAS** **:FETC:GSM:BLOC:PHAS?** The **FETC** command will return a data array, containing 711 floating point real numbers.

:FETCh:BLOCkdata:AFSPectrum?

Syntax	:FETCh:BLOCkdata:AFSPectrum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest data block describing the graph of the audio spectrum measurement. For further details regarding this measurement, please refer to the description of the :MEAS:BLOC:AFSP command.
Example	**:MEAS:BLOC:AFSP** **:FETC:BLOC:AFSP?** The **FETC** command will return a data array. The number of measurement result values delivered back is dependent on the span and resolution settings of the audio spectrum analyzer.

:FETCh:GSM:BLOCkdata:MSPectrum?

Syntax	:FETCh:GSM:BLOCkdata:MSPectrum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest data block describing the graph of the modulation spectrum measurement. For further details regarding this measurement, please refer to the description of the :MEAS:GSM:BLOC:MSP command.
Example	**:MEAS:GSM:BLOC:MSP** **:FETC:GSM:BLOC:MSP?** The **FETC** command will return a data array. The number of measurement result values delivered back is dependent on the span and resolution settings of the modulation spectrum analyzer.

:FETCh:BLOCkdata:PSCShape?

Syntax	:FETCh:BLOCkdata:PSCShape?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest data block describing the current consumption shape over a TDMA frame. The data block is a string consisting of 886 measurement result floating point real values with the physical dimension of mA. The current is sampled at a rate of 192 kHz, starting with the beginning of a new downlink TDMA frame.
Example	**:MEAS:GSM:CONT:BLOCkdata:PSCShape** **:FETCh:BLOC:PSCS?** Returns 886 measurement values, describing the current consumption shape over a TDMA frame.

:FETCh:GSM:RFRX:RBER:ALL?

Syntax	:FETCh:GSM:RFRX:RBER:ALL?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result values of the :MEAS:GSM[:ARRay]:RFRX:RBER:ALL measurement. The string returned will contain three floating point real numbers, separated by commas. The order of these numbers and their meaning is as follows: 1. the result of the residual bit error ratio measurement of the class Ib bits, 2. the result of the residual bit error ratio measurement of the class II bits, and 3. the result of the frame erasure ratio measurement.
Example	**:MEAS:GSM:RFRX:RBER:ALL** **:FETCh:GSM:RFRX:RBER:ALL?** This example will return the latest measurement result values for the three residual bit error measurements. The string delivered back is: **"0.2,1.7,0.3"**. This means that the RBER for class Ib bits is **0.2**.

:FETCh:GSM:RFRX:RBER:CIB?

Syntax	:FETCh:GSM:RFRX:RBER:CIB?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEAS:GSM[:ARRay]:RFRX:RBER:CIB measurement. The string delivered back will contain at least one floating point real number.
Example	**:MEAS:GSM:ARR:RFRX:RBER:CIB 5** **:FETCh:GSM:RFRX:RBER:CIB?** The string delivered back is: **"0.2,0.1,0.2,0.3,0.2"**.

:FETCh:GSM:RFRX:RBER:CII?

Syntax	:FETCh:GSM:RFRX:RBER:CII?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEAS:GSM[:ARRay]:RFRX:RBER:CII measurement. The string delivered back will contain at least one floating point real number.
Example	**:MEAS:GSM:ARR:RFRX:RBER:CII 5** **:FETCh:GSM:RFRX:RBER:CII?** The string delivered back is: **"1.5,1.7,2.1,2.3,2.6"**.

:FETCh:GSM:RFRX:RBER:FER?

Syntax	:FETCh:GSM:RFRX:RBER:FER?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEAS:GSM[:ARRay]:RFRX:RBER:FER measurement. The string delivered back will contain at least one floating point real number.
Example	**:MEAS:GSM:RFRX:RBER:FER** **:FETCh:GSM:RFRX:RBER:FER?** The string delivered back is: **"1.5"**.

:FETCh:GSM:RFRX:BER:ALL?

Syntax	:FETCh:GSM:RFRX:BER:ALL?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result values of the :MEAS:GSM[:ARRay]:RFRX:BER:ALL measurement. The string returned will contain three floating point real numbers, separated by commas. The order of these numbers and their meaning is as follows: 1. the result of the (nonresidual) bit error ratio measurement of the class Ia bits, 2. the result of the (nonresidual) bit error ratio measurement of the class Ib bits, and 3. the result of the (nonresidual) bit error ratio measurement of the class II bits.
Example	**:MEAS:GSM:ARR:RFRX:BER:ALL 3** **:FETCh:GSM:RFRX:BER:ALL?** This example will deliver back the latest measurement result values for the three residual bit error measurements. The string delivered back is: **"0.2,0.3,1.7,0.2,0.4,1.8,0.1,0.2,0.8"**.

:FETCh:GSM:RFRX:BER:CIA?

Syntax	:FETCh:GSM:RFRX:BER:CIA?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEAS:GSM[:ARRay]:RFRX:BER:CIA measurement. The string delivered back will contain at least one floating point real number.
Example	**:MEAS:GSM:RFRX:BER:CIA** **:FETCh:GSM:RFRX:BER:CIA?** The string delivered back is: **"0.1"**.

:FETCh:GSM:RFRX:BER:CIB?

Syntax	:FETCh:GSM:RFRX:BER:CIB?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEAS:GSM[:ARRay]:RFRX:BER:CIB measurement. The string delivered back will contain at least one floating point real number.
Example	**:MEAS:GSM:ARR:RFRX:BER:CIB 5** **:FETCh:GSM:RFRX:BER:CIB?** The string delivered back is: **"0.4,0.3,0.7,0.6,0.5"**.

:FETCh:GSM:RFRX:BER:CII?

Syntax	:FETCh:GSM:RFRX:BER:CII?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEAS:GSM[:ARRay]:RFRX:BER:CII measurement. The string delivered back will contain at least one floating point real number.
Example	**:MEAS:GSM:RFRX:BER:CII** **:FETCh:GSM:RFRX:BER:CII?** The string delivered back is: **"1.9"**.

:FETCh:GSM:RFRX:FBER?

Syntax	:FETCh:GSM:RFRX:FBER?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEAS:GSM[:ARRay]:RFRX:FBER measurement. The string delivered back will contain at least one floating point real number.
Example	**:MEAS:GSM:ARR:RFRX:FBER 5** **:FETCh:GSM:RFRX:FBER?** The string delivered back is: **"0.7,0.9,1.2,1.8,2.0"**.

:FETCh:GSM:RFRX:RTDelay?

Syntax	:FETCh:GSM:RFRX:RTDelay?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEASure:GSM:ARRay:RFRX:RTDelay measurement. The string delivered back will contain at least one integer.
Example	**:MEAS:GSM:RFRX:RTD** **:FETCh:GSM:RFRX:RTD?** The string delivered is: **"8"**.

:FETCh:GSM:RFSP:ACPM:MODulation?

Syntax	:FETCh:GSM:RFSP:ACPM:MODulation?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest data array for the ACPM modulation spectrum measurement. The string returned contains a multiple of 23 floating point real numbers, separated by commas. For further details regarding this measurement, please refer to the description of the :MEAS:GSM[:ARRay]:RFSPectrum:ACPM:MOD command.
Example	**:MEAS:GSM:RFSPectrum:ACPM:MOD** **:FETC:GSM:RFSP:ACPM:MOD?** The **FETC** command will return a data array, containing 23 floating point real numbers.

:FETCh:GSM:RFSPectrum:ACPM:MODPower?

Syntax	:FETCh[:GSM]:RFSPectrum:ACPM:MODPower?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest data array for the ACPM modulation spectrum measurement. The string returned contains a multiple of 24 floating point real numbers, separated by commas. For further details regarding this measurement, please refer to the description of the :MEAS:GSM:ARRay:RFSPectrum:ACPM:MODP command.
Example	:MEAS:RFSP:ACPM:MODP :FETC:RFSP:ACPM:MODP? The FETC command will return a data array, containing 24 floating point real numbers.

:FETCh:GSM:RFSP:ACPM:TRANsient?

Syntax	:FETCh:GSM:RFSP:ACPM:TRANsient?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command returns the latest data array for the measurement of the ACPM spectrum due to switching transients. The string returned contains a multiple of 27 floating point real numbers, separated by commas. For further details regarding this measurement, please refer to the description of the :MEAS:GSM[:ARRay]:RFSP:ACPM:TRAN command.
Example	**:MEAS:GSM:RFSPectrum:ACPM:TRAN** **:FETC:GSM:RFSP:ACPM:TRAN?** The **FETC** command will return a data array, containing 27 floating point real numbers.

:FETCh:AFANanlyser:GROup?

Syntax	:FETCh:AFANanlyser:GROup?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result values of the sequence of AF measurements as specified with the :CONF:MEAS:GRO:AFAN command and measured with the :MEAS[ARRay]:AFAN:GRO command. The string returned contains the related measurement result values, separated by commas. All measurement result values are floating point real numbers. The number of measurement result values handed back depends on the number and type of measurements defined with the **:CONF:MEAS:GRO:AFAN** command. The order of the measurement result values within the result string is as described below for the **:FETCh:AFANalyser:ALL** command.
Example	**:CONF:MEAS:GRO:AFAN SIN,FREQ** **:MEAS:AFAN:GRO** **:FETCh:AFAN:GRO?** In this example, the group of measurements is defined by a SINAD measurement combined with an audio frequency measurement. The values returned are: **"1000.0,53.5"**. Because of the internal order (see description of the command below), the first measurement result value delivered back is the audio frequency, the second one the SINAD.

:FETCh:AFANanlyser:ALL?

Syntax	:FETCh:AFANanlyser:ALL?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result values of the :MEAS[:ARRay]:AFAN:ALL measurement in a string. All measurement result values of the AF measurements will be floating point real numbers, separated by commas. The order of the measurement result values within the result string is as follows: 1. **ACVPeakp**, representing the AC peak-to-peak voltage of the AF signal, expressed in V(pp) 2. **ACVRms**, representing the rms-valued AC voltage of the AF signal, expressed in V(rms), 3. **DCVRms**, representing the rms-valued AC voltage on an applied DC signal, expressed in V(rms), 4. **FREQuency**, representing the audio frequency, expressed in Hertz, 5. **DISTortion**, representing the third-harmonic distortion of the applied sine-wave AF signal (expressed in percent), 6. **SINad**, representing the signal to noise ratio of the applied AF signal, expressed in dB. **Note:** Any AF test will need the audio option to be installed on your Willtek 3100.
Example	**:MEAS:ARR:AFAN:ALL 2** **:FETCh:AFAN:ALL?** In this example, the six most important AF measurements are performed twice in a sequence. The measurement result string in this example is: **"3.2,2.7,0.1,1000.0,1.1,53.5,** **2.7,1.7,0.08,1000.0,0.9,55.2"**.

:FETCh:AFAN:ACVoltage:PEAKp?

Syntax	:FETCh:AFAN:ACVoltage:PEAKp?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEAS[:ARRay]:AFAN:ACV:PEAKp measurement. The string delivered back will contain at least one floating point real number. **Note:** Any AF test will need the audio option to be installed on your Willtek 3100.
Example	**:MEAS:ARR:AFAN:ACV:PEAK 5** **:FETC:AFAN:ACV:PEAK?** The string delivered back is: **"3.2,3.1,3.3,3.2,2.9"**.

:FETCh:AFAN:ACVoltage:RMS?

Syntax	:FETCh:AFAN:ACVoltage:RMS?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	The query form of this command will return the latest measurement result value(s) of the `:MEAS[:ARRay]:AFAN:ACV:RMS` measurement. The string delivered back will contain at least one floating point real number. **Note:** Any AF test will need the audio option to be installed on your Willtek 3100.
Example	**`:MEAS:AFAN:ACV:RMS`** **`:FETC:AFAN:ACV:RMS?`** The string delivered back is: **`"1.7"`**.

`:FETCh:AFAN:DCVoltage?`

Syntax	`:FETCh:AFAN:DCVoltage?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the `:MEAS[:ARRay]:AFAN:DCV` measurement. The string delivered back will contain at least one floating point real number. **Note:** Any AF test will need the audio option to be installed on your Willtek 3100.
Example	**`:MEAS:AFAN:DCV`** **`:FETC:AFAN:DCV?`** The string delivered back is: **`"1.7"`**.

`:FETCh:AFANalyser:FREQuency?`

Syntax	`:FETCh:AFANalyser:FREQuency?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the `:MEAS[:ARRay]:AFAN:FREQ` measurement. The string delivered back will contain at least one floating point real number. **Note:** Any AF test will need the audio option to be installed on your Willtek 3100.
Example	**`:MEAS:ARR:AFAN:FREQ 5`** **`:FETC:AFAN:FREQ?`** The string delivered back is: **`"1000.5,1497.0,2004.7,2491.2,3005.0"`**.

`:FETCh:AFANalyser:DISTortion?`

Syntax	`:FETCh:AFANalyser:DISTortion?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the `:MEAS[:ARRay]:AFAN:DIST` measurement. The string delivered back will contain at least one floating point real number. **Note:** Any AF test will need the audio option to be installed on your Willtek 3100.

Example	**:MEAS:AFAN:DIST** **:FETC:AFAN:DIST?** The string delivered back is: **"1.5"**.

:FETCh:AFANalyser:SINad?

Syntax	:FETCh:AFANalyser:SINad?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command will return the latest measurement result value(s) of the :MEAS[:ARRay]:AFAN:SINad measurement. The string delivered back will contain at least one floating point real number. **Note:** Any AF test will need the audio option to be installed on your Willtek 3100.
Example	**:MEAS:ARR:AFAN:SIN 3** **:FETC:AFAN:SIN?** The string delivered back is: **"53.5,55.2,54.2"**.

:FETCh:PSUPply:GROup

Syntax	:FETCh:PSUPply:GROup?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This query returns the latest measurement result values of the sequence of power/current consumption measurements as specified with the :CONF:MEAS:GRO:PSUP command and measured with the :MEAS[ARRay]:PSUP:GRO command. The string returned contains the related measurement result values, separated by commas. All measurement result values are floating point real numbers. The number of measurement result values handed back depends on the number and type of measurements defined with the **:CONF:MEAS:GRO:PSUP** command. The order of the measurement result values within the result string is as described below for the **:FETCh:PSUP:ALL** command.
Example	**:CONF:MEAS:GRO:PSUP ACUR,APOW** **:MEAS:PSUP:GRO** **:FETCh:PSUP:GRO?** In this example, the group of measurements consists of the average current consumption and the average power consumption measurements. The values returned in this example are: **"859.2,293.7"**. Because of the internal order (see description of the command below), the first measurement result value delivered back is the power consumption (in mW), followed by the average current consumption (in mA).

:FETCh:PSUPply:ALL

Syntax	:FETCh:PSUPply:ALL?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This query returns the latest measurement result values of the :MEAS[:ARRay]:PSUP:ALL measurement in a string. All measurement result values of the power and current consumption measurements are floating point real numbers, separated by commas. The order of the measurement result values within the result string is as follows: 1. **APOW**, representing the average power consumption (in mW), 2. **ACUR**, representing the average current consumption, expressed in mA, 3. **PCUR**, representing the peak current consumption, expressed in mA. **Note:** Any current measurement will need the MS Power Supply and Current Measurement options to be installed on your Willtek 3100.
Example	**:MEAS:PSUP:ALL** **:FETCh:PSUP:ALL?** Returns a string like "**893.5,395.4,1256.2**".

:FETCh:PSUPply:APOWer

Syntax	:FETCh:PSUPply:APOWer?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This query returns the latest measurement result values of the :MEAS[:ARRay]:PSUP:APOW measurement in a string. The measurement result value is a floating point real numbers representing the average current consumption, expressed in mA. **Note:** Any current measurement will need the MS Power Supply and Current Measurement options to be installed on your Willtek 3100.
Example	**:MEAS:PSUP:APOW** **:FETCh:PSUP:APOW?** Returns a string like "**893.5**" giving the average power consumption in mW.

:FETCh:PSUPply:ACURrent

Syntax	:FETCh:PSUPply:ACURrent?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This query returns the latest measurement result values of the :MEAS[:ARRay]:PSUP:ACUR measurement in a string. The measurement result value is a floating point real numbers representing the average current consumption, expressed in mA. **Note:** Any current measurement will need the MS Power Supply and Current Measurement options to be installed on your Willtek 3100.

Example	**:MEAS:PSUP:ACUR** **:FETCh:PSUP:ACUR?** Returns a string like "**395.4**" giving the average current consumption in mA.

:FETCh:PSUPply:PCURrent

Syntax	:FETCh:PSUPply:PCURrent?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This query returns the latest measurement result values of the :MEAS[:ARRay]:PSUP:ALL measurement in a string. All measurement result values of the power and current consumption measurements are floating point real numbers, separated by commas. The order of the measurement result values within the result string is as follows: 1. **APOW**, representing the average power consumption (in mW), 2. **ACUR**, representing the average current consumption, expressed in mA, 3. **PCUR**, representing the peak current consumption, expressed in mA. **Note:** Any current measurement will need the MS Power Supply and Current Measurement options to be installed on your Willtek 3100.
Example	**:MEAS:PSUP:PCUR** **:FETCh:PSUP:PCUR?** Returns a string like "**1256.2**" giving the peak current consumption in mA.

:FETCh:WCDMa:LAST

Syntax	:FETCh:WCDMa:LAST
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Returns exactly the same values returned by the last FETCh: or MEASure: query.
Example	:FETC:WCDM:LAST?

:FETCh:WCDMa:RFTX:FREQ:ERRor

Syntax	:FETCh:WCDMa:RFTX:FREQ:ERRor?
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the results of the :MEASure:WCDMa:CONT|ARRay:RFTX:FREQ:ERR measurement command.
Example	:FETC:WCDM:RFTX:FREQ? Returns the measured frequency error, e.g. –7.834.

:FETCh:WCDMa:RFTX:POWer:MEAN

Syntax	`:FETCh:WCDMa:RFTX:POWer:MEAN?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The query form returns the latest result value of the `MEASure:WCDMa:RFTX:POWer:MEAN` measurement respectively the latest result values of the `MEASure:WCDMa:ARRay:RFTX:POWer:MEAN` measurement. The values delivered will contain at least one floating-point number in dBm.
Example	`:FETC:WCDM:RFTX:POW:MEAN?` Returns an RF power measurement result, e.g. `-45.3`.

:FETCh:WCDMa:RFTX:POWer:PEAK

Syntax	`:FETCh:WCDMa:RFTX:POWer:PEAK?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The query form returns the result of a previous `:MEASure:WCDMa:CONT|ARRay:RFTX:POWer:PEAK` measurement. The values delivered will contain at least one floating-point number in dBm.
Example	`:FETC:WCDM:RFTX:POW:PEAK?` Returns the Peak RF power measurement result, e.g. `-36.8`.

:FETCh:WCDMa:RFTX:NSTair[:DATa]

Syntax	`:FETCh:WCDMa:RFTX:NSTair[:DATa]?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The query form of this command returns the latest result value of the `MEASure:WCDMa:CONT|ARRay:RFTX:NSTair` measurement. The values delivered will contain 501 floating-point number in dBm.
Example	`:FETC:WCDM:RFTX:NST?`

:FETCh:WCDMa:RFTX:NSTair:POWer

Syntax	`:FETCh:WCDMa:RFTX:NSTair:POWer?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The query form of this command returns the latest result value of the `MEASure:WCDMa:CONT|ARRay:RFTX:NSTair:POWer` measurement. The values delivered will contain at least seven floating-point numbers in dBm.

Example	`:FETC:WCDM:RFTX:NST:POW?`

`:FETCh:WCDMa:RFTX:ZSPan:POWer`

Syntax	`:FETCh:WCDMa:RFTX:ZSPan:POWer?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The query form of this command returns the latest result value of the `MEASure:WCDMa:RFTX:ZSPan:POWer` measurement. The result delivered will contain as many floating point numbers in dBm as defined with the `RFAN:WCDM:ZSP:REFP` command.
Example	`:FETC:WCDM:RFTX:ZSP:POW?-54.9,-62.7,-61.3.`

`:FETCh:WCDMa:RFTX:CPOWer`

Syntax	`:FETCh:WCDMa:RFTX:CPOWer?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the results of the `:MEASure:WCDMa:CONT|ARRay:RFTX:CHPower` measurement command.
Example	`:FETC:WCDM:RFTX:CPOW?`

`:FETCh:WCDMa:RFTX:MODQuality:ALL?`

Syntax	`:FETCh:WCDMa:RFTX:MODQuality:ALL?`
Parameters	There are no parameters.
Description	Only the query form is supported.

Query	The command returns the result of a previous :MEASure:WCDMa:CONT|ARR:RFTX:ALL measurement command. It delivers a string, containing 10 measurement result values, separated by commas. The order and type of these measurement result values delivered back is as follows: 1. EVM RMS, floating point real number representing the RMS vector error measurement in percent, 2. EVM Peak, floating point real number representing the peak vector error measurement in percent, 3. Magnitude error RMS, floating point real number representing the RMS magnitude vector error measurement in percent, 4. Magnitude error Peak, floating point real number representing the PEAK magnitude vector error measurement in percent, 5. Phase error RMS, floating point real number representing the RMS phase vector error measurement in degrees, 6. Phase error Peak, floating point real number representing the PEAK phase vector error measurement in degrees, 7. Frequency error, floating point real number representing the mobile's frequency error, 8. RHO, floating point real number representing the mobile's modulation quality, 9. I/Q Offset, a floating point value with the result in dBc, representing the result of the origin offset vector error measurement, 10. I/Q Imbalance, floating point real number representing the result of the IQ Imbalance vector error measurement with the result in dB. **Note:** For a further description of the single measurements, see description of the related commands below.
Example	:MEAS:WCDMA:RFTX:MODQ:ALL :FETC:WCDM:RFTX:MODQ:ALL? In this case, all relevant RF TX measurements will be performed in a sequence, e.g. 21.624,72.8,15.335,72.736,6.738,27.512,-6.378,0.961,-28.251,-43.743.

:FETCh:WCDMa:RFTX:MODQuality:ERMS?

Syntax	:FETCh:WCDMa:RFTX:MODQuality:ERMS?
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous :MEASure:WCDMa:CONT|ARR:RFTX:MODQuality:ERMS measurement command.
Example	:FETC:WCDM:RFTX:MODQ:ERMS? Returns the measured EVM RMS vector error, e.g. 21.624.

FETCh:WCDMa:RFTX:MODQuality:EPEAk?

Syntax	:FETCh:WCDMa:RFTX:MODQuality:EPEAk?
Parameters	There are no parameters.
Description	Only the query form is supported.

Query	The command returns the result of a previous `:MEASure:WCDMa:CONT|ARR:RFTX:MODQuality:EPEAk` measurement command.
Example	`:FETC:WCDM:RFTX:MODQ:EPEAk?` Returns the measured EVM peak vector error, e.g. `72.8`.

`:FETCh:WCDMa:RFTX:MODQuality:MRMS?`

Syntax	`:FETCh:WCDMa:RFTX:MODQuality:MRMS?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT|ARR:RFTX:MODQuality:MRMS` measurement command.
Example	`:FETC:WCDM:RFTX:MODQuality:MRMS?` Returns the measured Magnitude RMS vector error, e.g. `15.335`.

`:FETCh:WCDMa:RFTX:VERRor:MODQuality:MPEAk?`

Syntax	`:FETCh:WCDMa:RFTX:VERRor:MODQuality:MPEAk?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT|ARR:RFTX:MODQuality:MPEAk` measurement command.
Example	`:FETC:WCDM:RFTX:MODQ:MPEA?` Returns the measured magnitude peak vector error, e.g. `72.736`.

`:FETCh:WCDMa:RFTX:MODQuality:PRMS?`

Syntax	`:FETCh:WCDMa:RFTX:MODQuality:PRMS?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT|ARR:RFTX:MODQuality:PRMS` measurement command.
Example	`:FETC:WCDM:RFTX:MODQ:PRMS?` Returns the measured RMS phase vector error, e.g. `6.738`.

`:FETCh:WCDMa:RFTX:MODQuality:PPEAk?`

Syntax	`:FETCh:WCDMa:RFTX:MODQuality:PPEAk?`

Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT|ARR:RFTX:MODQuality:PPEAk` measurement command.
Example	`:FETC:WCDM:RFTX:MODQ:PPEAk?` Returns the measured phase peak vector error, e.g. `27.512`.

:FETCh:WCDMa:RFTX:MODQuality:RHO?

Syntax	`:FETCh:WCDMa:RFTX:MODQuality:RHO?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT|ARR:RFTX:MODQuality:RHO` measurement command.
Example	`:FETC:WCDM:RFTX:MODQ:RHO?` Returns the modulation quality, e.g. `0.9989`.

:FETCh:WCDMa:RFTX:MODQuality:IQOFfset?

Syntax	`:FETCh:WCDMa:RFTX:MODQuality:IQOFfset?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT|ARR:RFTX:MODQuality:IQOFfset` measurement command.
Example	`:FETC:WCDM:RFTX:MODQ:IQOF?` Returns the origin offset in dBc, e.g. `-28.251`.

:FETCh:WCDMa:RFTX:MODQuality:IQIMbalance?

Syntax	`:FETCh:WCDMa:RFTX:MODQuality:IQIMbalance?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT|ARR:RFTX:MODQuality:IQIMbalance` measurement command.
Example	`:FETC:WCDM:RFTX:MODQ:IQIM?` Returns the IQ imbalance in dBc, e.g. `-43.743`.

:FETCh:WCDMa:RFSPectrum:MSPectrum[:DATa]?

Syntax	`:FETCh:WCDMa:RFSPectrum:MSPectrum[:DATa]?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:RFSPectrum:MSPectrum[:DATA|AVG]` measurement command.
Example	`MEAS:WCDM:RFSP:MSP:AVG 10 :FETC:WCDM:RFSP:MSP?`

:FETCh:WCDMa:RFSPectrum:MSPectrum:OBW?

Syntax	`:FETCh:WCDMa:RFSPectrum:MSPectrum:OBW?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT|ARR:RFSPectrum:MSPectrum:OBW` measurement command.
Example	`:FETC:WCDM:RFSP:MSP:OBW?`

:FETCh:WCDMa:RFSPectrum:ACLR?

Syntax	`:FETCh:WCDMa:RFSPectrum:ACLR?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The query form of this command returns the latest result value of the `MEAS:WCDM:RFSP:ACLR` measurement. The values delivered will contain at least one array of floating-point numbers in dBm. The results for the adjacent channel leakage power ratio at the following offsets: -10 MHz, -5 MHz, 0 MHz, +5 MHz, +10 MHz.

Example	`:MEAS:WCDMA:ARR:RFSP:ACLR 5`
	`:FETC:WCDM:RFSP:ACLR?`
	The `Fetch` command will return a data array, containing 5 times 5 floating point real numbers.

:FETCh:WCDMa:RFSPectrum:SEM[:DATa]?

Syntax	`:FETCh:WCDMa:RFSPectrum:SEM[:DATa]?`
Parameters	There are no parameters.
Description	Only the query form is supported.

Query	The command returns the result of a previous `:MEASure:WCDMa:RFSPectrum:SEM[:DATA|AVG]` Spectrum Emission Mask measurement.
Example	`MEAS:WCDM:RFSP:SEM:AVG 10 :FETC:WCDM:RFSP:SEM?`

:FETCh:WCDMa:RFRX:BER:RATIo?

Syntax	`:FETCh:WCDMa:RFRX:BER:RATIo?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT:RFRX:BER:RATIo` measurement command.
Example	`:FETC:WCDM:RFRX:BER?`

:FETCh:WCDMa:RFRX:BLER:RATIo?

Syntax	`:FETCh:WCDMa:RFRX:BLER:RATIo?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	The command returns the result of a previous `:MEASure:WCDMa:CONT:RFRX:BLER:RATIo` measurement command.
Example	`:FETC:WCDM:RFRX:BLER?`

:FETCh:WCDMa:UEReport:ALL

Syntax	`:FETCh:WCDMa:UEReport:ALL`
Parameters	There are no parameters.
Description	Only the query form is supported.

Query	The query delivers a string, containing 6 measurement result values, separated by commas started by `MEASure:WCDMa[:CONT|ARRay]:UEReport:ALL`. The order and type of these measurement result values delivered is as follows: 1. `CPICh ECNO`, floating point real number representing UE measured normalized chip energy per noise value of the P-CPICH, 2. `CPICh RSCP`, floating point real number representing UE measured received signal code power of the P-CPICH, 3. `PLOSs`, floating point real number representing the pathloss between Node B and UE, estimated by the UE, 4. `TXPower`, floating point real number representing the UE transmit power, 5. `RTTDifference`, floating point real number representing UE measured RX-Time-Difference between UL and DL signal, 6. `TRCH BLER`, floating point real number representing the UE estimated BLER value for the Transport Channel defined with `CONFigure:WCDMa:MEASure:UEReport:TRCH`. **Note:** For a further description of the single measurements, see description of the related commands below.

Example	`:MEAS:WCDM:UER:ALL?`
	`:FETC:WCDM:UER:ALL?`
	All UE Report measurements will be performed in a sequence, e.g. `23,85,102,97,992,3`.

:FETCh:WCDMa:UEReport:CPICh:ECNO

Syntax	`:FETCh:WCDMa:UEReport:CPICh:ECNO?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Reports the UE measured normalized chip energy per noise value of the P-CPICH started by `MEASure:WCDMa[:CONT|ARRay]:UEReport:CPICh:ECNO`.
Example	`:FETC:WCDM:UER:CPICh:ECNO?` Starts the measurement and returns the result, e.g. `22`.

:FETCh:WCDMa:UEReport:CPICh:RSCP

Syntax	`:FETCh:WCDMa:UEReport:CPICh:RSCP?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Reports the UE measured received signal code power of the P-CPICH started by `MEASure:WCDMa[:CONT|ARRay]:UEReport:CPICh:RSCP`.
Example	`:FETC:WCDM:UER:CPICh:RSCP?` Starts the measurement and returns the result, e.g. `74`.

:FETCh:WCDMa:UEReport:PLOSs

Syntax	`:FETCh:WCDMa:UEReport:PLOSs?`
Parameters	There are no parameters.

Description	The query form is supported only, started by `MEASure:WCDMa[:CONT|ARRay]:UEReport:PLOSs`.
Query	Reports the path loss between Node B and UE, estimated by the UE.
Example	`:FETC:WCDM:UER:PLOS?` Starts the measurement and returns the result, e.g. `85`.

:FETCh:WCDMa:UEReport:TXPOwer

Syntax	`:FETCh:WCDMa:UEReport:TXPOwer?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Reports the UE transmit power started by `MEASure:WCDMa[:CONT|ARRay]:UEReport:TXPower`.
Example	`:FETC:WCDM:UER:TXPO?` Starts the measurement and returns the result, e.g. `66`.

:FETCh:WCDMa:UEReport:RTTDifference

Syntax	`:FETCh:WCDMa:UEReport:RTTDifference?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Reports the UE measured RX-Time-Difference between UL and DL signal started by `MEASure:WCDMa[:CONT|ARRay]:UEReport:RTTDifference`.
Example	`:FETC:WCDM:UER:RTTD?` Starts the measurement and returns the result, e.g. `956`.

:FETCh:WCDMa:UEReport:TRCH:BLER

Syntax	`:FETCh:WCDMa:UEReport:TRCH:BLER?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Reports the UE estimated BLER value for the Transport Channel started by `MEASure:WCDMa[:CONT|ARRay]:UEReport:TRCH:BLER`.
Example	`:FETC:WCDM:UER:TRCH:BLER?` Starts the measurement and returns the result, e.g. `0`.

CALCulate Subsystem

The CALCulate subsystem provides a large number of commands in order to set limits, check measurement result values against those limits and to perform statistic evaluation of measurement result values.

:CALCulate:RESet

Syntax	:CALCulate:RESet
Parameters	There are no parameters.
Description	Resets the **CALC** subsystem and brings it into a defined operating state. We kindly recommend to use this command to initialize the **CALC** subsystem when starting a new test run.
Query	There is no query form of this command available.
Example	**:CALCulate:RESet** **:MEAS:GSM:ARR:RFTX:LENG 10** **:CALC:MAVerage?** This example first resets the **CALC** subsystem and then starts the measurement of the frequency error for 10 measurement runs. The average measurement result value of those ten measurement runs will then be calculated and returned by the **:CALC:MAV?** query (see below for details regarding this command).

:CALCulate:LIMit:FAIL[:LAST]?

Syntax	:CALCulate:LIMit:FAIL[:LAST]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Checks whether the result(s) of the **last** measurement performed did violate the user-defined measurement result limits or not. As long as all single measurement results are within the limits, a **0** will be returned. A **1** indicates that at least one measurement result did violate at least one limit. The limits for the single measurements can be set using the appropriate commands of the **CALC** subsystem (as explained in this section).
Example	**:CALCulate:RESet** **:MEAS:GSM:RFTX:ALL** **:CALC:LIM:FAIL?** This example first resets the **CALC** subsystem and then starts the measurement of all relevant RF TX parameters. If a measurement result of the current sequence violates the corresponding measurement result limits, the query will return a **1**. When all measurement results are within their limits, a **0** will be returned.

:CALCulate:LIMit:FAIL:CUMulative?

Syntax	:CALCulate:LIMit:FAIL:CUMulative?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	Checks whether **any** result(s) of the measurement started last did violate their corresponding, user-defined measurement result limits. As long as all single measurement results are within their limits, a **0** will be returned. A **1** indicates that at least one measurement result did violate at least one limit. The limits for the single measurements can be set using the appropriate commands of the **CALC** subsystem (as explained in this section). To reset the cumulative check of the measurement results, use the **:CALC:LIM:FAIL:CUM:RES** command as explained below.
Example	**:CALC:LIM:FAIL:CUM:RES** **:MEAS:GSM:RFTX:TEMP** **...(other SCPI commands)... :CALC:LIM:FAIL:CUM?** This example first resets the cumulative process then starts the continuous check of the UL burst against the power/time template (PTT). After that, other SCPI commands are performed. After a while, the **:CALC:LIM:FAIL:CUM** command is used to check whether there has been any violation of the PTT since the start of the **:MEAS:GSM:RFTX:TEMP** measurement.

:CALCulate:LIMit:FAIL:CUMulative:RESet

Syntax	:CALCulate:LIMit:FAIL:CUMulative:RESet
Parameters	There are no parameters.
Description	This command resets the cumulative process as explained above (see description of command **:CALC:LIM:FAIL:CUM** for details).
Query	There is no query form of this command available.
Example	**:CALC:LIM:FAIL:CUM:RES** **:MEAS:GSM:RFTX:TEMP** **...(other SCPI commands)... :CALC:LIM:FAIL:CUM?** This example first resets the cumulative process then starts the continuous check of the UL burst against the power/time template (PTT). After that, other SCPI commands are performed. After a while, the **:CALC:LIM:FAIL:CUM** command is used to check whether there has been any violation of the PTT since the start of the **:MEAS:GSM:RFTX:TEMP** measurement.

:CALCulate:MAVerage?

Syntax	:CALCulate:MAVerage?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Calculates and returns the **average** measurement result value of the measurement started last. The string delivered back will contain as many average values as measurement types performed. The single average values will always have the format of floating point real numbers and will be separated by commas.

Example	**:CONF:MEAS:GRO:RFTX POW,PRMS** **:CALC:RES** **:MEAS:GSM:ARR:RFTX:GROup 20** **:CALCulate:MAVerage?** In this example, first the group of measurements is defined by a power measurement combined with a RMS phase error measurement. This group of measurements is performed 20 times. After that, the single measurement result values will be averaged and returned. The values returned in this example are: **"4.53,9.98"**.

:CALCulate:MMINimum?

Syntax	:CALCulate:MMINimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **minimum** measurement result value of the measurement started last. The string delivered back will contain as many minimum values as measurement types performed. The single minimum values will always have the format of floating point real numbers and will be separated by commas.
Example	**:MEAS:AFAN:SIN** **:CALC:RES** **...(other SCPI commands)...** **:CALCulate:MMIN?** In this example, first a continuous SINAD measurement is started. After some time, the **:CALC:MMIN** command is used to read out the minimum SINAD measured. The string returned in this example is **"42.6"**.

:CALCulate:MMAXimum?

Syntax	:CALCulate:MMAXimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **maximum** measurement result value of the measurement started last. The string delivered back will contain as many maximum values as measurement types performed. The single maximum values will always have the format of floating point real numbers and will be separated by commas.
Example	**:MEAS:GSM:RFTX:FREQ** **:CALC:RES** **...(other SCPI commands)...** **:CALCulate:MMAX?** In this example, first a continuous measurement of the mobile's frequency error is started. After some time, the **:CALC:MMAX** command is used to read out the maximum frequency error of the mobile. The string returned in this example is **"22.1"**.

:CALCulate:MSIGma?

Syntax	:CALCulate:MSIGma?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **mean value** and the **standard deviation** of the measurement results since the last measurement start. The string delivered back contains as many pairs of mean and standard deviation values as measurement types performed. The individual values all have the format of floating point real numbers and will be separated by commas.
Example	:MEAS:AFAN:SIN :CALC:RES ...(other SCPI commands)... :CALCulate:MMIN? In this example, first a continuous SINAD measurement is started. After some time, the :CALC:MMIN command is used to read out the minimum SINAD measured. The string returned in this example is "42.6".

:CALCulate:GSM:RFTX:MAVerage?

Syntax	:CALCulate:GSM:RFTX:MAVerage?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Calculates and returns the **average** measurement result value of the TX measurement started last. The string delivered back will contain as many average values as measurement types performed. The single average values will always have the format of floating point real numbers and will be separated by commas.
Example	:CONF:GSM:MEAS:GRO:RFTX POW,PRMS :CALC:RES :MEAS:GSM:ARR:RFTX:GRO 20 :CALC:GSM:RFTX:MAV? In this example, first the group of measurements is defined by a power measurement combined with a RMS phase error measurement. This group of measurements is performed 20 times. After that, the single measurement result values will be averaged and returned. The values returned in this example are: "4.53,9.98".

:CALCulate:GSM:RFTX:MMINimum?

Syntax	:CALCulate:GSM:RFTX:MMINimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	Returns the **minimum** measurement result value of the TX measurement started last. The string delivered back will contain as many minimum values as measurement types performed. The single minimum values will always have the format of floating point real numbers and will be separated by commas.

Example	`:CONF:GSM:MEAS:GRO:RFTX POW,PRMS`
	`:CALC:RES`
	`:MEAS:GSM:ARR:RFTX:GRO 20`
	`:CALC:GSM:RFTX:MMIN?`
	In this example, first the group of measurements is defined by a power measurement combined with a RMS phase error measurement. This group of measurements is performed 20 times. After that, the single measurement result values will be averaged and returned. The values returned in this example are:
	`"4.53,9.98"`.
	Because of the internal order (see description of the `:MEAS:GSM[:ARR]:RFTX:ALL` command, the first measurement result value delivered back is the minimum RMS phase error, the second one the mobile's minimum RF output power.

:CALCulate:GSM:RFTX:MMINimum?

Syntax	`:CALCulate:GSM:RFTX:MMINimum?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **minimum** measurement result value of the TX measurement started last. The string delivered back will contain as many minimum values as measurement types performed. The single minimum values will always have the format of floating point real numbers and will be separated by commas.

Example	`:CONF:GSM:MEAS:GRO:RFTX POW,PRMS`
	`:CALC:RES`
	`:MEAS:GSM:ARR:RFTX:GRO 20`
	`:CALC:GSM:RFTX:MMIN?`
	In this example, first the group of measurements is defined by a power measurement combined with a RMS phase error measurement. This group of measurements is performed 20 times. After that, the single measurement result values will be averaged and returned. The values returned in this example are:
	`"4.53,9.98"`.
	Because of the internal order (see description of the `:MEAS:GSM[:ARR]:RFTX:ALL` command, the first measurement result value delivered back is the minimum RMS phase error, the second one the mobile's minimum RF output power.

:CALCulate:GSM:RFTX:MMAXimum?

Syntax	`:CALCulate:GSM:RFTX:MMAXimum?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	Returns the **maximum** measurement result value of the TX measurement started last. The string delivered back will contain as many maximum values as measurement types performed. The single maximum values will always have the format of floating point real numbers and will be separated by commas.

Example	`:MEAS:GSM:RFTX:FREQ`
	`:CALC:RES`
	`...(other SCPI commands)...`
	`:CALC:GSM:RFTX:MMAX?`
	In this example, first a continuous measurement of the mobile's frequency error is started. After some time, the `:CALC:GSM:RFTX:MMAX` command is used to read out the maximum frequency deviation of the mobile. The string returned in this example is `"22.1"`.

:CALCulate:GSM:RFTX:ALL:LIMit[:FAIL]?

Syntax	`:CALCulate:GSM:RFTX:ALL:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers 19 boolean numbers in a string, separated by commas. These numbers indicate, whether the single measurement results of the `:MEAS:GSM[:ARRay]:RFTX:ALL` measurement did violate their test limits. The position of the boolean number within the string returned indicates the result of the limit check for the following measurements: 1. peak phase error measurement, 2. root-mean square phase error measurement , 3. measurement of the mobile's frequency error, 4. measurement of the burst lenght, 5. timing error measurement, 6. measurement of the mobile's RF output power level, 7. check of the burst against the PTT, 8...15. measurement of RF power level at the corner points, 16. measurement of the absolute minimum RF power level during the active part of the burst, 17. the position of this minimum, 18, the measurement of the absolute maximum RF power level during the active part of the burst, and 19. the position of this maximum. **Note:** No limits can be set for parameters 16...19 in this software release. Therefore, positions 16...19 in the string delivered back will be set to **0** by definition.
Example	**`:CALC:GSM:RFTX:ALL:LIM?`** String returned: **`"0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0"`** The **1** on position 6 indicates that the mobile's RF power level violated its test limits.

:CALCulate:GSM:RFTX:ALL:LIMit:STATe

Syntax	`:CALCulate:GSM:RFTX:ALL:LIMit:STATe <PredefExp>`
Parameters	**`PredefExp`** is one of the following predefined expressions: **`ON|OFF`**. Default is **`ON`**.

Description	This command switches the limit check for the relevant RF TX measurements either on or off (see description of the `:MEASure:GSM:RFTX:ALL` command for reference).
Query	There is no query form of this command available.
Example	**`:CALC:GSM:RFTX:ALL:LIM:STAT OFF`** Switches the limit check for the most relevant RF TX measurements off.

:CALCulate:GSM :RFTX:ALL:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GSM`
	`:RFTX:ALL:LIMit:UPPer[:DATA]`
	`<real1>,<real2>,<real3>,<real4>, <real5>,<real6>`

Parameters	**realx** are six floating point real numbers. **real1**: The minimum value is **0.0**, the maximum is **90.0**. The minimum resolution possible is **0.1**, the default value is **20.0**. **real2**: The minimum value is **0.0**, the maximum is **90.0**. The minimum resolution possible is **0.1**, the default value is **5.0**. **real3**: The minimum value is **0.0**, the maximum is **100,000.0**. The minimum resolution possible is **1.0**, the default value is **90.0**. **real4**: The minimum value is **0.0**, the maximum is **700.0**. The minimum resolution possible is **0.1**, the default value is **562.8**. **real5**: The minimum value is **0.0**, the maximum is **64.0**. The minimum resolution possible is **0.01**, the default value is **3.0**. **real6**: The minimum value is **0.0**, the maximum is **99.0**. The minimum resolution possible is **0.01**, the default value is **2.0**.
Description	Sets the **upper** limits for six measurement types with one command. The order of the floating point real numbers defines their meaning: **real1** represents the upper limit of the peak phase error measurement; the physical dimension of the number stated is degrees. **real2** represents the upper limit of the RMS-valued phase error measurement; the physical dimension of the number stated is degrees. **real3** represents the upper limit of the frequency error measurement; the physical dimension of the number stated is Hertz. **real4** represents the upper limit of the measurement of the burst length; the physical dimension of the number stated is microseconds. **real5** represents the upper limit of the measurement of the uplink timing error (with the ordered timing advance being taken into account). The physical dimension of the number stated is microseconds. **real6** represents the upper limit of the RF power level measurement; the physical dimension of the number stated is dBm.
Query	There is no query form of this command available.
Example	**`:CALC:GSM:RFTX:ALL:LIM:UPP 20,5,90,562.8,0,13`** The command sets the upper limits for the transmitter measurement to "20.0, 5.0, 90.0, 562.8, 0.0, 13.0", which could be the limits for the next RFTX:ALL measurements.

:CALCulate:GSM :RFTX:ALL:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFTX:ALL:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>,<real4>, <real5>,<real6>
Parameters	**realx** are six floating point real numbers. **real1**: The minimum value is **-90.0**, the maximum is **0.0**. The minimum resolution possible is **0.1**, the default value is **-20.0**. **real2**: The minimum value is **-90.0**, the maximum is **0.0**. The minimum resolution possible is **0.1**, the default value is **-5.0**. **real3**: The minimum value is **-100,000.0**, the maximum is **0.0**. The minimum resolution possible is **1.0**, the default value is **-90.0**. **real4**: The minimum value is **0.0**, the maximum is **700.0**. The minimum resolution possible is **0.1**, the default value is **542.8**. **real5**: The minimum value is **-64.0**, the maximum is **0.0**. The minimum resolution possible is **0.01**, the default value is **-3.0**. **real6**: The minimum value is **-99.0**, the maximum is **0.0**. The minimum resolution possible is **0.01**, the default value is **-3.0**.
Description	Sets the **lower** limits for six measurement types with one command. The order of the floating point real numbers defines their meaning: **real1** represents the lower limit of the peak phase error measurement; the physical dimension of the number stated is degrees. **real2** represents the lower limit of the RMS-valued phase error measurement; the physical dimension of the number stated is degrees. **real3** represents the lower limit of the frequency error measurement; the physical dimension of the number stated is Hertz. **real4** represents the lower limit of the measurement of the burst length; the physical dimension of the number stated is microseconds. **real5** represents the lower limit of the measurement of the uplink timing error (with the ordered timing advance being taken into account). The physical dimension of the number stated is microseconds. **real6** represents the lower limit of the RF power level measurement; the physical dimension of the number stated is dBm.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:ALL:LIM:LOW 0,0,-90,542.8,0,10** The command sets the lower limits for the transmitter measurement to **"0.0,0.0,-90.0,542.8,0.0,10.0"**, which could be the limits for the next RFTX:ALL measurements.

:CALCulate:GSM:RFTX:PPEAk:LIMit[:FAIL]?

Syntax	:CALC:GSM:RFTX:PPEAk:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	This command delivers one boolean number, indicating whether the user-definable limits of the peak phase error measurement were violated by a measurement result value or not. A **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:MEAS:GSM:ARRay:PPEAK 20** **:CALC:GSM:RFTX:PPEAk:LIM?** String returned:**"1"** This result indicates that at least one measurement result value did violate the limits of the peak phase error measurement.

:CALCulate:GSM:RFTX:PPEAk:LIMit:STATe

Syntax	:CALC:GSM:RFTX:PPEAk:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFTX:PPEAk measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:PPEAk:LIM:STAT OFF** Switches the limit check for the peak phase error measurement off.

:CALCulate:GSM:RFTX:PPEAk:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM:RFTX:PPEAk:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **90.0**. The minimum resolution possible is **0.1**, the default value is **20.0**.
Description	Sets the **upper** limit of the peak phase error measurement; the physical dimension of the number stated is degrees.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:PPEA:LIM:UPP 15** Sets the upper limit for the peak phase error to 15 degrees.

:CALCulate:GSM:RFTX:PPEAk:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM:RFTX:PPEAk:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **-90.0**, the maximum is **0.0**. The minimum resolution possible is **0.1**, the default value is **-20.0**.
Description	Sets the **lower** limit of the peak phase error measurement; the physical dimension of the number stated is degrees.
Query	There is no query form of this command available.

Example	**:CALC:GSM:RFTX:PPEA:LIM:LOW -15** Sets the lower limit for the peak phase error measurement to -15 degrees.

:CALCulate:GSM:RFTX:PRMS:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFTX:PRMS:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the RMS-valued phase error measurement were violated by a measurement result value or not. A **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:MEAS:GSM:ARRay:RFTX:PRMS 10** **:CALC:GSM:RFTX:PRMS:LIM?** String returned:**"0"** This string delivered back indicates that there was no violation of the measurement limits.

:CALCulate:GSM:RFTX:PRMS:LIMit:STATe

Syntax	:CALCulate:GSM:RFTX:PRMS:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFTX:PRMS measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:PRMS:LIM:STAT OFF** Switches the limit check for the RMS-valued phase error measurement off.

:CALCulate:GSM:RFTX:PRMS:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM:RFTX:PRMS:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **90.0**. The minimum resolution possible is **0.1**, the default value is **5.0**.
Description	Sets the **upper** limit of the RMS-valued phase error measurement; the physical dimension of the number stated is degrees.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:PRMS:LIM:UPP 6.2** Sets the upper limit for the RMS-valued phase error measurement to 6.2 degrees.

:CALCulate:GSM:RFTX:PRMS:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM:RFTX:PRMS:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **-90.0**, the maximum is **0.0**. The minimum resolution possible is **0.1**, the default value is **-5.0**.
Description	Sets the **lower** limit of the RMS-valued phase error measurement; the physical dimension of the number stated is degrees.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:PRMS:LIM:LOW 0.0** Sets the lower limit for the RMS-valued phase error measurement to 0 degrees.

::CALCulate:GSM:RFTX:FREQuency:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFTX:FREQuency:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the frequency error measurement were violated by a measurement result value or not. A **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:MEAS:GSM:ARRay:RFTX:FREQ 20** **:CALC:GSM:RFTX:FREQ:LIM?** String returned:**"0"** This string delivered back indicates that there was no violation of the measurement limits.

CALCulate:GSM:RFTX:FREQuency:LIMit:STATe

Syntax	:CALCulate:GSM:RFTX:FREQuency:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFTX:FREQ measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:FREQ:LIM:STAT ON** Switches the limit check for the frequency error measurement on.

:CALCulate:GSM:RFTX:FREQuency:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM:RFTX:FREQuency:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100000.0**. The minimum resolution possible is **1.0**, the default value is **90.0**.
Description	Sets the **upper** limit of the frequency error for measurements in asynchronous mode (Gen/Ana); the physical dimension of the number stated is Hertz.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:FREQ:LIM:UPP 45** Sets the upper limit of the frequency error measurement result to 45 Hz.

:CALCulate:GSM:RFTX:FREQuency:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM:RFTX:FREQuency:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **-100000.0**, the maximum is **0.0**. The minimum resolution possible is **1.0**, the default value is **-90.0**.
Description	Sets the **lower** limit of the frequency error for measurements in asynchronous mode; the physical dimension of the number stated is Hertz.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:FREQ:LIM:LOW -60.8** Sets the lower limit of the frequency error measurement result to -60.8 Hz.

:CALCulate:GSM:RFTX:FREQuency:LIMit:GSM

Syntax	:CALCulate:GSM:RFTX:FREQuency:LIMit:GSM <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100000.0**. The minimum resolution possible is **1.0**, the default value is **90.0**.
Description	Sets the **symmetrical** limits of the frequency error for measurements in synchronous (call) mode for GSM-900 and GSM-850; the physical dimension of the number stated is Hertz.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:FREQ:LIM:GSM 65** Sets the lower limit of the frequency error measurement result on GSM-900 and GSM-850 channels to -65 Hz and the upper limit to +65 Hz.

:CALCulate:GSM:RFTX:FREQuency:LIMit:PCN

Syntax	:CALCulate:GSM:RFTX:FREQuency:LIMit:PCN <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100000.0**. The minimum resolution possible is **1.0**, the default value is **180.0**.

Description Sets the **symmetrical** limits of the frequency error for measurements in synchronous (call) mode for GSM-1800 (also known as DCS-1800 or PCN); the physical dimension of the number stated is Hertz.

Query There is no query form of this command available.

Example **:CALC:GSM:RFTX:FREQ:LIM:PCN 120**
Sets the lower limit of the frequency error measurement result on GSM-1800 channels to -120 Hz and the upper limit to +120 Hz.

:CALCulate:GSM:RFTX:FREQuency:LIMit:PCS

Syntax :CALCulate:GSM:RFTX:FREQuency:LIMit:PCS <real1>

Parameters **real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100000.0**. The minimum resolution possible is **1.0**, the default value is **180.0**.

Description Sets the **symmetrical** limits of the frequency error for measurements in synchronous (call) mode for GSM-1900 (also known as PCS); the physical dimension of the number stated is Hertz.

Query There is no query form of this command available.

Example **:CALC:GSM:RFTX:FREQ:LIM:PCS 120**
Sets the lower limit of the frequency error measurement result on GSM-1900 channels to -120 Hz and the upper limit to +120 Hz.

:CALCulate:GSM:RFTX:LENGth:LIMit[:FAIL]?

Syntax :CALCulate:GSM:RFTX:LENGth:LIMit[:FAIL]?

Parameters There are no parameters.

Description There is solely a query form of this command available.

Query This command delivers one boolean number, indicating whether the user-definable limits of the burst length measurement were violated by a measurement result value or not.
A **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits.
The limits can be set using the commands described below.

Example **:MEAS:GSM:ARRay:RFTX:LENG 10**
:CALC:GSM:RFTX:LENG:LIM?
String returned:**"0"**
This string indicates that there was no violation of the measurement limits.

:CALCulate:GSM:RFTX:LENGth:LIMit:STATe

Syntax :CALCulate:GSM:RFTX:LENGth:LIMit:STATe <PredefExp>

Parameters **PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.

Description This command switches the check of the measurement result values of the MEAS:GSM:RFTX:LENGth measurement against their user-definable limits either on or off.

Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:LENG:LIM:STAT ON** Switches the limit check for the burst length measurement on.

:CALCulate:GSM:RFTX:LENGth:LIMit:UPPer[:DATA]

Syntax	:CALC:GSM:RFTX:LENG:LIM:UPP[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **700.0**. The minimum resolution possible is **0.1**, the default value is **562.8**.
Description	Sets the **upper** limit of the burst length measurement; the physical dimension of the number stated is microseconds.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:LENG:LIM:UPP 563** Sets the upper limit of the burst length to 563.0 µs.

:CALCulate:GSM:RFTX:LENGth:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM:RFTX:LENGth:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **700.0**. The minimum resolution possible is **0.1**, the default value is **542.8**.
Description	Sets the **lower** limit of the burst length measurement; the physical dimension of the number stated is microseconds.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:LENG:LIM:LOW 541.9** Sets the lower limit of the burst length to 541.9 µs.

:CALCulate:GSM:RFTX:UTIMe:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFTX:UTIMe:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the UL timing error measurement were violated by a measurement result value or not. A **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:MEAS:GSM:ARRay:RFTX:UTIM 10** **:CALC:GSM:RFTX:UTIM:LIM?** String returned:**"0"** This string indicates that there was no violation of the measurement limits.

:CALCulate:GSM:RFTX:UTIMe:LIMit:STATe

Syntax	:CALCulate:GSM:RFTX:UTIMe:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFTX:UTIMe measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:UTIM:LIM:STAT ON** Switches the limit check for the UL timing error measurement on.

:CALCulate:GSM:RFTX:UTIMe:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM:RFTX:UTIMe:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **64.0**. The minimum resolution possible is **0.01**, the default value is **3**.
Description	Sets the **upper** limit of the UL timing error measurement; the physical dimension of the number stated is microseconds. **Note:** The UL timing error measurement takes the ordered timing advance into account.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:UTIM:LIM:UPP 5** Sets the upper limit of the uplink timing error measurement result to 5.0 µs.

:CALCulate:GSM:RFTX:UTIMe:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM:RFTX:UTIMe:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **-64.0**, the maximum is **0.0**. The minimum resolution possible is **0.01**, the default value is **-3**.
Description	Sets the **lower** limit of the UL timing error measurement; the physical dimension of the number stated is microseconds. **Note:** The UL timing error measurement takes the ordered timing advance into account.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:UTIM:LIM:LOW -5** Sets the lower limit of the uplink timing error measurement result to -5.0 µs.

:CALCulate:GSM:RFTX:POWer:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFTX:POWer:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	This command delivers one boolean number, indicating whether the user-definable limits of the measurement of the mobile's RF output power level were violated by a measurement result value or not. A **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:MEAS:GSM:ARRay:POW 20** **:CALC:GSM:RFTX:POW:LIM?** String returned:**"0"** This string indicates that there was no violation of the measurement limits.

:CALCulate:GSM:RFTX:POWer:LIMit:STATe

Syntax	:CALCulate:GSM:RFTX:POWer:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFTX:POWer measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:POW:LIM:STAT ON** Switches the limit check for the RF output power level measurement on.

:CALCulate:GSM:RFTX:POWer:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM:RFTX:POWer:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **99.0**. The minimum resolution possible is **0.01**, the default value is **2**.
Description	Sets the **upper** limit for the mobile's RF output power level. The physical dimension of the number stated is dBm.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:POW:LIM:UPP 13** Sets the upper limit of the peak power measurement to 13 dBm.

:CALCulate:GSM:RFTX:POWer:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM:RFTX:POWer:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **-99.0**, the maximum is **0.0**. The minimum resolution possible is **0.01**, the default value is **-2**.
Description	Sets the **lower** limit for the mobile's RF output power level. The physical dimension of the number stated is dBm.
Query	There is no query form of this command available.

Example	**:CALC:GSM:RFTX:POW:LIM:LOW -45** Sets the lower limit of the peak power measurement to -45 dBm.

:CALCulate:GSM:RFTX:POWer:LIMit:GSM

Syntax	:CALCulate:GSM:RFTX:POWer:LIMit:GSM <real1>,<real2>,...,<real18>
Parameters	**real1** through **real18** are floating point real numbers. The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.1**. The default values are: **2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5**.
Description	Sets the **symmetrical** limits of the peak power for measurements in synchronous (call) mode on GSM-900 and GSM-850 channels; the physical dimension of the number stated is dB. The first value applies to power control steps 0-2 (39 dBm), the second value applies to power control 3 (37 dBm) and so forth up to the eighteenth value for power control steps 19 through 31 (5 dBm).
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:POW:LIM:GSM 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5** Sets the lower and upper limits of the power measurement result on GSM-900 and GSM-850 channels to 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5 dB around the nominal level.

:CALCulate:GSM:RFTX:POWer:LIMit:PCN

Syntax	:CALCulate:GSM:RFTX:POWer:LIMit:PCN <real1>,<real2>,...,<real19>
Parameters	**real1** through **real19** are floating point real numbers. The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.1**. The default values are: **2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5**.
Description	Sets the **symmetrical** limits of the peak power for measurements in synchronous (call) mode on GSM-1800 channels; the physical dimension of the number stated is dB. The first value applies to power control step 29 (36 dBm), the second value applies to power control 30 (34 dBm) and so forth up to the nineteenth value for power control steps 15 through 28 (0 dBm).
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:POW:LIM:PCN 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5** Sets the lower and upper limits of the power measurement result on GSM-1800 channels to 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5 dB around the nominal level.

:CALCulate:GSM:RFTX:POWer:LIMit:PCS

Syntax	:CALCulate:GSM:RFTX:POWer:LIMit:PCS <real1>,<real2>,...,<real18>
Parameters	**real1** through **real19** are floating point real numbers. The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.1**. The default values are: **2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5**.
Description	Sets the **symmetrical** limits of the peak power for measurements in synchronous (call) mode on GSM-1900 channels; the physical dimension of the number stated is dB. The first value applies to power control step 30 (34 dBm), the second value applies to power control step 31 (32 dBm), the third value applies to power control step 0 (30 dBm) and so forth up to the eighteenth value for power control steps 15 through 28 (0 dBm).
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:POW:LIM:PCN 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5** Sets the lower and upper limits of the power measurement result on GSM-1800 channels to 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5 dB around the nominal level.

:CALCulate:GSM:RFTX:TEMPlate[:NORMal]:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFTX:TEMPlate[:NORMal]:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers 14 boolean numbers, indicating whether the PTT limits for a **normal** burst were violated or not. A **0** on a position means that there was no violation of the limits while a **1** indicates that at least one measurement result did violate one of the corresponding measurement limits. The 14 boolean numbers correspond to the limits of the PTT as defined on the user interface or with the **:CALC:GSM:RFTX:TEMP** commands as described below. The first eight numbers stand for the eight 'upper' segments of the burst, while the following six numbers stand for the six 'lower' segments of the burst. Refer to section Power/Time Templates (PTT) for further details.
Example	**:MEAS:GSM:RFTX:TEMP** **:CALC:GSM:RFTX:TEMP:LIM?** String returned:**"0,0,0,0,0,0,0,0,0,0,1,0,0,0"** This string indicates that there was a violation of the limits of one of the 'lower' segments of the PTT (in this example during the ramping up phase of the burst).

:CALCulate:GSM:RFTX:TEMPlate[:NORMal]:LIMit:AUTO

Syntax	:CALCulate:GSM:RFTX:TEMPlate[:NORMal]:LIMit:AUTO <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.

Description	This command selects the power/time template to check normal bursts against. **Auto mode** (parameter **ON**) means that the Willtek 3100 will use the appropriate PTT as defined in the GSM/ETSI standards. Those PTTs vary with the frequency range and the RF power level used by the mobile. **Manual mode** (parameter **OFF**) means that the GSM/ETSI specifications will be ignored and that the Willtek 3100 will use the manually set limits instead (see description of the **:CALC:GSM:RFTX:TEMP:LIM:UPP** and **:CALC:GSM:RFTX:TEMP:LIM:LOW** commands for further details).
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:TEMP:LIM:AUTO ON** Switches the limit check for the burst shape to use the standard PTTs as defined by ETSI standards.

:CALCulate:GSM:RFTX:TEMPlate[:NORMal]:LIMit:STATe

Syntax	:CALCulate:GSM:RFTX:TEMPlate[:NORMal]:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFTX:TEMPlate measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:TEMP:LIM:STAT ON** Switches on the limit check of the burst shape against the PTT for normal bursts.

:CALCulate:GSM :RFTX:TEMPlate[:NORMal]:LIMit:UPPer[:DATA]

Syntax

```
:CALCulate:GSM
:RFTX:TEMPlate[:NORMal]:LIMit:UPPer[:DATA]
<real0t>,<real0p>,<real1t>,<real1p>,
<real2t>,<real2p>,<real3t>,<real3p>,
<real4t>,<real4p>,<real5t>,<real5p>,
<real6t>,<real6p>,<real7t>,<real7p>, <real8t>,<real8p>
```

Parameters

realxt and **realxp** are floating point real numbers.
The minimum value for all **realxt** is **-41.0**, the maximum value **580.0**. The minimum resolution possible for all **realxt** is **0.1**. The default values are:
for **real0t**: **-41.0**,
for **real1t**: **-28.0**,
for **real2t**: **-18.0**,
for **real3t**: **-10.0**
for **real4t**: **0.0**,
for **real5t**: **553.0**
for **real6t**: **561.0**,
for **real7t**: **571.0**
for **real8t**: **580.0**.
The minimum value for all **realxp** is **-150.0**, the maximum value **5.0**. The minimum resolution possible for all **realxp** is **0.1**. The default values are:
for **real0p**: **-59.0**,
for **real1p**: **-59.0**,
for **real2p**: **-30.0**,
for **real3p**: **-6.0**
for **real4p**: **4.0**,
for **real5p**: **1.0**
for **real6p**: **-6.0**,
for **real7p**: **-30.0**
for **real8p**: **-59.0**.

Description

Sets the **upper** limits for the user-definable PTT (for normal bursts). The upper limits of the user-defined PTT are made up by nine pairs of position (in microseconds relative to the beginning of the burst) and the related RF power level (in dB relative to the power level during the active part of the burst).
For further details regarding the user-definable PTT, refer to section Power/Time Templates (PTT).

Query

The query form of this command is not available.

Example

```
:CALC:GSM:RFTX:TEMP:LIM:UPP -41,-47,-28,-47,-18,-28,-
10,-4,0 ,4,553,1,561,-4,571,-28,580,-47
```

:CALCulate:GSM :RFTX:TEMPlate[:NORMal]:LIMit:LOWer[:DATA]

Syntax

```
:CALCulate:GSM
:RFTX:TEMPlate[:NORMal]:LIMit:LOWer[:DATA]
<real0t>,<real0p>,<real1t>,<real1p>,
<real2t>,<real2p>,<real3t>,<real3p>,
<real4t>,<real4p>,<real5t>,<real5p>, <real6t>,<real6p>
```

Parameters	**realxt** and **realxp** are floating point real numbers. The minimum value for all **realxt** is **-41.0**, the maximum value **580.0**. The minimum resolution possible for all **realxt** is **0.1**. The default values are: for **real0t**: **-41.0**, for **real1t**: **0.0**, for **real2t**: **0.0**, for **real3t**: **20.0** for **real4t**: **270.0**, for **real5t**: **543.0** for **real6t**: **543.0**. The minimum value for all **realxp** is **-150.0**, the maximum value **5.0**. The minimum resolution possible for all **realxp** is **0.1**. The default values are: for **real0p**: **-150.0**, for **real1p**: **-150.0**, for **real2p**: **-40.0**, for **real3p**: **-1.0** for **real4p**: **-1.0**, for **real5p**: **-1.0** for **real6p**: **-150.0**.
Description	Sets the **lower** limits for the user-definable PTT (for normal bursts). The lower limits of the user-defined PTT are made up by seven pairs of position (in microseconds relative to the beginning of the burst) and the related RF power level (in dB relative to the power level during the active part of the burst). For further details regarding the user-definable PTT, refer to section Power/Time Templates (PTT).
Query	The query form of this command is not available.
Example	:CALC:GSM:RFTX:TEMP:LIM:LOW -41,-150,0,-150,0,-40,20, -1,270,-1,543,-1,543,-150

:CALCulate:GSM :RFTX:TEMPlate:RACH:LIMit[:FAIL]?

Syntax	:CALCulate:GSM :RFTX:TEMPlate:RACH:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers 14 boolean numbers, indicating whether the PTT limits for a **random access** burst were violated or not. A **0** on a position means that there was no violation of the limits while a **1** indicates that at least one measurement result did violate one of the corresponding measurement limits. The 14 boolean numbers correspond to the limits of the PTT as defined on the user interface or with the **:CALC:GSM:RFTX:TEMP:RACH:LIM** commands as described below. The first eight numbers stand for the eight 'upper' segments of the burst, while the following six numbers stand for the six 'lower' segments of the burst. Rrefer to section Power/Time Templates (PTT) for further details.
Example	**:CALC:GSM:RFTX:TEMP:RACH:LIM?** String returned:**"0,0,0,0,0,0,0,0,0,0,1,0,0,0"** This string indicates that there was a violation of the limits of one of the 'lower' segments of the PTT (in this example during the ramping up phase of the burst).

:CALCulate:GSM:RFTX:TEMPlate:RACH:LIMit:AUTO

Syntax	:CALCulate:GSM:RFTX:TEMPlate:RACH:LIMit:AUTO <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command selects the power/time template to check a random access burst against. **Auto mode** (parameter **ON**) means that the Willtek 3100 will use the appropriate PTT as defined in the GSM/ETSI standards. Those PTTs vary with the frequency range and the RF power level used by the mobile. **Manual mode** (parameter **OFF**) means that the GSM/ETSI specifications will be ignored and that the Willtek 3100 will use the manually set limits instead (see description of the **:CALC:GSM:RFTX:TEMP:RACH:LIM:UPP** and **:CALC:GSM:RFTX:TEMP:RACH:LIM:LOW** commands for further details).
Query	There is on query form to this commmand.
Example	**:CALC:GSM:RFTX:TEMP:RACH:LIM:AUTO ON** Switches the limit check for the shape of **random access** bursts to use the standard PTTs as defined by GSM/ETSI.

:CALCulate:GSM:RFTX:TEMPlate:RACH:LIMit:STATe

Syntax	:CALCulate:GSM:RFTX:TEMPlate:RACH:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the limit check of the burst shape against the PTT (of random access bursts) either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:TEMP:RACH:LIM:STAT ON** Switches on the limit check of the shape of **random access** bursts against the PTT.

:CALCulate:GSM
:RFTX:TEMPlate:RACH:LIMit:UPPer[:DATA]

Syntax	:CALC:GSM :RFTX:TEMPlate:RACH:LIMit:UPPer[:DATA] <real0t>,<real0p>,<real1t>,<real1p>, <real2t>,<real2p>,<real3t>,<real3p>, <real4t>,<real4p>,<real5t>,<real5p>, <real6t>,<real6p>,<real7t>,<real7p>, <real8t>,<real8p>
Parameters	**realxt** and **realxp** are floating point real numbers. The minimum value for all **realxt** is **-41.0**, the maximum value **580.0**. The minimum resolution possible for all **realxt** is **0.1**. The default values are: for **real0t**: **-41.0**, for **real1t**: **-28.0**, for **real2t**: **-18.0**, for **real3t**: **-10.0** for **real4t**: **0.0**, for **real5t**: **331.0** for **real6t**: **339.0**, for **real7t**: **349.0** for **real8t**: **349.0**. The minimum value for all **realxp** is **-150.0**, the maximum value **5.0**. The minimum resolution possible for all **realxp** is **0.1**. The default values are: for **real0p**: **-59.0**, for **real1p**: **-59.0**, for **real2p**: **-30.0**, for **real3p**: **-6.0** for **real4p**: **4.0**, for **real5p**: **1.0** for **real6p**: **-6.0**, for **real7p**: **-30.0** for **real8p**: **-59.0**.
Description	Sets the **upper** limits for the user-definable PTT for random access bursts. The upper limits of the user-defined PTT are made up by nine pairs of position (in microseconds relative to the beginning of the burst) and the related RF power level (in dB relative to the power level during the active part of the burst). For further details regarding the user-definable PTT, refer to section Power/Time Templates (PTT).
Query	The query form of this command is not supported.
Example	:CALC:GSM:RFTX:TEMP:RACH:LIM:UPP -41,-59,-28,-59,-18,-30,-10,-6,0, 4,331,1,339,-6,349,-30,349,-59

:CALCulate:GSM :RFTX:TEMPlate:RACH:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFTX:TEMPlate:RACH:LIMit:LOWer[:DATA] <real0t>,<real0p>,<real1t>,<real1p>, <real2t>,<real2p>,<real3t>,<real3p>, <real4t>,<real4p>,<real5t>,<real5p>, <real6t>,<real6p>
Parameters	**realxt** and **realxp** are floating point real numbers. The minimum value for all **realxt** is **-41.0**, the maximum value **580.0**. The minimum resolution possible for all **realxt** is **0.1**. The default values are: for **real0t**: **-41.0**, for **real1t**: **0.0**, for **real2t**: **0.0**, for **real3t**: **20.0** for **real4t**: **270.0**, for **real5t**: **321.0** for **real6t**: **321.0**. The minimum value for all **realxp** is **-150.0**, the maximum value **5.0**. The minimum resolution possible for all **realxp** is **0.1**. The default values are: for **real0p**: **-150.0**, for **real1p**: **-150.0**, for **real2p**: **-40.0**, for **real3p**: **-1.0** for **real4p**: **-1.0**, for **real5p**: **-1.0** for **real6p**: **-150.0**.
Description	Sets the **lower** limits for the user-definable PTT for random access bursts. The lower limits of the user-defined PTT are made up by seven pairs of position (in microseconds relative to the beginning of the burst) and the related RF power level (in dB relative to the power level during the active part of the burst). For further details regarding the user-definable PTT, refer to section Power/Time Templates (PTT).
Query	The query form of this command is not supported.
Example	:CALC:GSM:RFTX:TEMP:RACH:LIM:LOW -41,-150,0,-150,0,-40,20, -1,270,-1,321,-1,321,-150

:CALCulate:GSM :RFTX:CORNer[:NORMal]:POSition

Syntax	:CALCulate:GSM :RFTX:CORNer[:NORMal]:POSition <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>
Parameters	**realx** are floating point real numbers. The minimum value for all **realx** is **-28.0**, the maximum value **580.0**. The minimum resolution possible for all **realx** is **0.1**. The default values are: for **real1**: **-28.0**, for **real2**: **-18.0**, for **real3**: **-10.0**, for **real4**: **0.0** for **real5**: **542.8**, for **real6**: **552.8** for **real7**: **560.8**. for **real8**: **570.8**.
Description	Sets the position of the eight user-definable corner points in relation to the beginning of the burst for **normal** bursts. The positions are given in microseconds. For further details, refer to section Corner Points.
Query	There is no query form to this command.
Example	:CALC:GSM:RFTX:CORN:POS -28,-18,- 10,0,542.8,552.8,562.8,572.8

:CALCulate:GSM :RFTX:CORNer[:NORMal]:LIMit[:FAIL]?

Syntax	:CALCulate:GSM :RFTX:CORNer[:NORMal]:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers eight boolean numbers, indicating whether the limits for the eight corner positions of **normal** bursts were violated by the bursts received or not. A **0** on a position means that there was no violation of the limits while a **1** indicates that at least one measurement result did violate the corresponding measurement limits. The eight boolean numbers correspond with the eight corner points set with the **:CALC:GSM:RFTX:CORN:NORM:POS** command as described above. Please refer to section Corner Points for further details.
Example	:MEAS:GSM:ARR:CORN 10 :CALC:GSM:RFTX:CORN:LIM:FAIL? String returned in this example:"0,0,1,0,0,0,0,0" This string indicates that the limits set for corner point 3 were violated.

:CALCulate:GSM :RFTX:CORNer[:NORMal]:LIMit:STATe

Syntax	:CALCulate:GSM :RFTX:CORNer[:NORMal]:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFTX:CORNer measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	:CALC:GSM:RFTX:CORN:LIM:STAT OFF Switches the limit check at the eight corner positions off.

:CALCulate:GSM :RFTX:CORNer[:NORMal]:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM :RFTX:CORNer[:NORMal]:LIMit:UPPer[:DATA] <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>
Parameters	**realx** are floating point real numbers. The minimum value for all **realx** is **-150.0**, the maximum is **10.0**. The minimum resolution possible for all **realx** is **0.01**. The default value is **4.0** for all **realx**.
Description	Sets the **upper** limits at the eight user-definable corner points for **normal** bursts. The single data values are dB relative to the power level during the active part of the burst.
Query	There is no query form to this command.
Example	:CALC:GSM:RFTX:CORN:LIM:UPP 4,4,4,4,4,4,4,4

:CALCulate:GSM :RFTX:CORN[:NORM]:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFTX:CORN[:NORM]:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>
Parameters	**realx** are floating point real numbers. The minimum value for all **realx** is **-150.0**, the maximum is **10.0**. The minimum resolution possible for all **realx** is **0.01**. The default value is **-150.0** for all **realx**.
Description	Sets the **lower** limits at the eight user-definable corner points for **normal** bursts. The single data values are dB relative to the power level during the active part of the burst.
Query	There is no query form to this command.

Example	:CALC:GSM:RFTX:CORN:LIM:LOW -150,-150,-150,-150,-150,-150,-150,-150

:CALCulate:GSM:RFTX:CORNer:RACH:POSition

Syntax	:CALCulate:GSM:RFTX:CORNer:RACH:POSition <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>
Parameters	**realx** are floating point real numbers. The minimum value for all **realx** is **-28.0**, the maximum value **580.0**. The minimum resolution possible for all **realx** is **0.1**. The default values are: for **real1**: **-28.0**, for **real2**: **-18.0**, for **real3**: **-10.0**, for **real4**: **0.0** for **real5**: **321.2**, for **real6**: **331.2** for **real7**: **339.2**. for **real8**: **349.2**.
Description	Sets the position of the eight user-definable corner points in relation to the beginning of the burst for **random access** bursts. The positions are given in microseconds. For further details, please refer to section Corner Points.
Query	There is no query form to this command.
Example	:CALC:GSM:RFTX:CORN:RACH:POS -28,-18,-10,0,321.2,331.2,341.2,351.2

:CALC:GSM
:RFTX:CORNer:RACH:LIMit[:FAIL]?

Syntax	:CALC:GSM :RFTX:CORNer:RACH:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers eight boolean numbers, indicating whether the limits for the eight corner positions of **random access** burst were violated by the bursts received or not. A **0** on a position means that there was no violation of the limits while a **1** indicates that at least one measurement result did violate one of the two corresponding measurement limits. The eight boolean numbers correspond with the eight corner points set with the **:CALC:GSM:RFTX:CORN:RACH:POS** command as described above.
Example	**:CALC:GSM:RFTX:CORN:RACH:LIM:FAIL?** String returned:**"0,1,0,0,0,0,0,0"** This string indicates that one of the two limits set for corner point 2 was violated.

:CALCulate:GSM:RFTX:CORN:RACH:LIMit:STATe

Syntax	:CALCulate:GSM:RFTX:CORN:RACH:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the bursts against the limits at the corner points either on or off for **random access** bursts.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:CORN:RACH:LIM:STAT OFF** Switches the limit check at the eight corner positions off.

:CALCulate:GSM :RFTX:CORNer:RACH:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM :RFTX:CORNer:RACH:LIMit:UPPer[:DATA] <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>
Parameters	**realx** are floating point real numbers. The minimum value for all **realx** is **-150.0**, the maximum is **10.0**. The minimum resolution possible for all **realx** is **0.01**. The default value is **4.0** for all **realx**.
Description	Sets the **upper** limits at the eight user-definable corner points for **random access** bursts. The single data values are dB relative to the power level during the active part of the burst.
Query	There is no query form to this command.
Example	:CALC:GSM:RFTX:CORNer:NORMal:LIM:UPP:DATA 4,4,4,4,4,4,4,4

:CALCulate:GSM :RFTX:CORNer:RACH:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFTX:CORNer:RACH:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>
Parameters	**realx** are floating point real numbers. The minimum value for all **realx** is **-150.0**, the maximum is **10.0**. The minimum resolution possible for all **realx** is **0.01**. The default value is **-150.0** for all **realx**.
Description	Sets the **lower** limits at the eight user-definable corner points for **random access** bursts. The single data values are dB relative to the power level during the active part of the burst.
Query	There is no query form to this command.
Example	:CALC:GSM:RFTX:CORN:RACH:LIM:LOW -150,-150,-150,-150,-150,-150,-150,-150

:CALCulate:GSM:RFRX:MAVerage?

Syntax	:CALCulate:GSM:RFRX:MAVerage?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Calculates and returns the **average** measurement result value of the RX measurement started last. The string delivered back will contain as many average values as measurement types performed. The single average values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:GSM:ARR:RFRX:ALL** **:CALC:GSM:RFRX:MAV?**

:CALCulate:GSM:RFRX:MMINimum?

Syntax	:CALCulate:GSM:RFRX:MMINimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **minimum** measurement result value of the RX measurement started last. The string delivered back will contain as many minimum values as measurement types performed. The single minimum values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:GSM:ARR:RFRX:ALL** **:CALC:GSM:RFRX:MMIN?**

:CALCulate:GSM:RFRX:MMAXimum?

Syntax	:CALCulate:GSM:RFRX:MMAXimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **maximum** measurement result value of the RX measurement started last. The string delivered back will contain as many maximum values as measurement types performed. The single maximum values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:GSM:ARR:RFRX:ALL** **:CALC:GSM:RFRX:MMAX?**

:CALCulate:GSM:RFRX:MSIGma?

Syntax	:CALCulate:GSM:RFRX:MSIGma?
Parameters	There are no parameters.

Description There is solely a query form of this command available.

Query Returns the **mean value** and the **standard deviation** of the RX measurement last started. The string delivered back contains as many pairs of mean and standard deviation values as measurement types performed. The individual values all have the format of floating point real numbers and will be separated by commas.

Example
```
:CALC:RES
:MEAS:GSM:RFRX:RBER:CII
...(other SCPI commands)...
:CALC:GSM:RFRX:MSIG?
```
In this example, first a continuous measurement of the residual bit error rate (RBER) on class II bits is started. After some time, the **:CALC:GSM:RFRX:MSIG** command is used to read out the mean RBER and its standard deviation. The string returned in this example is "**5.3, 3.1**".

:CALCulate:GSM :RFRX:RBER:ALL:LIMit[:FAIL]?

Syntax
```
:CALCulate:GSM
:RFRX:RBER:ALL:LIMit[:FAIL]?
```

Parameters There are no parameters.

Description There is solely a query form of this command available.

Query This command delivers 3 boolean numbers in a string, separated by commas. These numbers indicate, whether one of the residual bit error ratio measurements did violate the corresponding test limits. While a **0** delivered back on any position means that the related test was passed without exceeding the limits, a **1** indicates that at least one of the limits was violated by the measurement result value.
The position of the boolean number within the string returned indicates the result of the check for the following measurments:
1. the bit error ratio measurement of the class 1b bits,
2. the bit error ratio measurement of the class II bits and,
3. the frame erasure ratio measurement.

Notes
- For further details regarding bit classes, please refer to section voice coding.
- For details regarding the three measurement types, see the description of the :MEAS:GSM:RFRX:RBER:ALL command.

Example
```
:MEAS:GSM:RFRX:RBER:ALL
:CALC:GSM:RFRX:RBER:ALL:LIM?
```
String returned:**"0,0,1"**
The **1** on position 3 indicates that there was a violation of at least one limit of the frame erasure measurement.

:CALCulate:GSM:RFRX:RBER:ALL:LIMit:STATe

Syntax
```
:CALCulate:GSM:RFRX:RBER:ALL:LIMit:STATe <PredefExp>
```

Parameters **PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.

Description This command switches the check of the measurement result values of the MEAS:GSM:RFRX:RBER:ALL measurement against their user-definable limits either on or off.

Query	There is no query form of this command available.
Example	**:CALC:GSM:RFRX:RBER:ALL:LIM:STAT OFF** Switches the limit check of all residual bit error ratio measurements off.

:CALCulate:GSM :RFRX:RBER:ALL:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM :RFRX:RBER:ALL:LIMit:UPPer[:DATA] <real1>,<real2>,<real3>
Parameters	**realx** are three floating point real numbers. The minimum value for all **realx** is **0.0**, the maximum is **100.0**. The minimum resolution possible for all **realx** is **0.1**. The default value for all **realx** is **2.5**.
Description	Sets the **upper** limits for three residual bit error ratio measurements with one command. The order of the floating point real numbers defines their meaning: **real1** represents the upper limit of the bit error ratio measurement for the class 1b bits, **real2** represents the upper limit of the bit error ratio measurement of the class II bits, while **real3** represents the upper limit of the frame erasure ratio measurement. **Note:** For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not available.
Example	**:CALC:GSM:RFRX:RBER:ALL:LIM:UPP 1,2,1** Sets the upper limits as follows: – RBER class Ia: 1.0% – RBER class II: 2.0% – FER: 1.0%

:CALCulate:GSM :RFRX:RBER:ALL:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFRX:RBER:ALL:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>
Parameters	**realx** are three floating point real numbers. The minimum value for all **realx** is **0.0**, the maximum is **100.0**. The minimum resolution possible for all **realx** is **0.1**. The default value for all **realx** is **0.0**.
Description	Sets the **lower** limits for three residual bit error ratio measurements with one command. The order of the floating point real numbers defines their meaning: **real1** represents the lower limit of the bit error ratio measurement for the class 1b bits, **real2** represents the lower limit of the bit error ratio measurement of the class II bits, while **real3** represents the lower limit of the frame erasure ratio measurement.
Query	The query form of this command is not available.
Example	**:CALC:GSM:RFRX:RBER:ALL:LIM:LOW 0,0,0,** Sets the lower limits for RBER class Ia bits, RBER class II bits and FER to 0.0%.

:CALCulate:GSM :RFRX:RBER:CIB:LIMit[:FAIL]?

Syntax	:CALCulate:GSM :RFRX:RBER:CIB:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating, whether the result value of the residual bit error ratio measurement of the **class Ib** bits did violate the user-definable test limits. While a **0** delivered back means that the related test was passed without exceeding the limits, a **1** indicates that at least one limit was violated by the measurement result value.
Example	**:MEAS:GSM:RFRX:RBER:CIB** **:CALC:GSM:RFRX:RBER:CIB:LIM?** String returned:**"0"** This means that the bit error ratio measured was within the limits set with the commands explained in this section.

:CALCulate:GSM:RFRX:RBER:CIB:LIMit:STATe

Syntax	:CALCulate:GSM:RFRX:RBER:CIB:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFRX:RBER:CIB measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFRX:RBER:CIB:LIM:STAT OFF** Switches the limit check of the residual bit error ratio measurement for the class 1b bits off.

:CALCulate:GSM :RFRX:RBER:CIB:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM :RFRX:RBER:CIB:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **2.5**.
Description	Sets the **upper** limit for the residual bit error ratio measurement of the **class Ib** bits.
Query	The query form of this command is not available.
Example	**:CALC:GSM:RFRX:RBER:CIB:LIM:UPP 5** Sets the upper limit for the class I bit error rate results to 5.0.

:CALCulate:GSM :RFRX:RBER:CIB:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFRX:RBER:CIB:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.
Description	Sets the **lower** limit for the residual bit error ratio measurement of the **class Ib** bits.
Query	The query form of this command is not available.
Example	**:CALC:GSM:RFRX:RBER:CIB:LIM:LOW 0.0** Sets the lower limit for the class I bit error rate results to 0.

:CALCulate:GSM :RFRX:RBER:CII:LIMit[:FAIL]?

Syntax	:CALCulate:GSM :RFRX:RBER:CII:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating, whether the result value of the residual bit error ratio measurement of the **class II** bits did violate the user-definable test limits. While a **0** delivered back means that the related test was passed without exceeding the limits, a **1** indicates that at least one limit was violated by the measurement result value.
Example	**:MEAS:GSM:RFRX:RBER:CII** **:CALC:GSM:RFRX:RBER:CII:LIM?** String returned:**"1"** This means that the bit error ratio measured was outside the limits set with the commands explained in this section.

:CALCulate:GSM:RFRX:RBER:CII:LIMit:STATe

Syntax	:CALCulate:GSM:RFRX:RBER:CII:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFRX:RBER:CII measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFRX:RBER:CII:LIM:STAT ON** Switches the limit check of the residual bit error ratio measurement for the class II bits on.

:CALCulate:GSM
:RFRX:RBER:CII:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM :RFRX:RBER:CII:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **2.5**.
Description	Sets the **upper** limit for the residual bit error ratio measurement of the **class II** bits.
Query	The query form of this command is not available.
Example	**:CALC:GSM:RFRX:RBER:CII:LIM:UPP 5** Sets the upper limit of the class II RBER measurement to 5%.

:CALCulate:GSM
:RFRX:RBER:CII:LIMit:LOWer[:DATA]

Syntax	:CALC:GSM :RFRX:RBER:CII:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.
Description	Sets the **lower** limit for the residual bit error ratio measurement of the **class II** bits.
Query	The query form of this command is not available.
Example	**:CALC:GSM:RFRX:RBER:CII:LIM:LOW 0.0** Sets the lower limit of the class II RBER measurement to 0%.

:CALCulate:GSM:RFRX:RBER:FER:LIM[:FAIL]?

Syntax	:CALCulate:GSM:RFRX:RBER:FER:LIM[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating, whether the result value of the frame erasure ratio measurement did violate the user-definable test limits. While a **0** delivered back means that the related test was passed without exceeding the limits, a **1** indicates that at least one limit was violated by the measurement result value. **Note:** For further details regarding the frame error ratio, please refer to section RBER live display.
Example	**:MEAS:GSM:RFRX:RBER:FER** **:CALC:GSM:RFRX:RBER:FER:LIM?** String returned:**"0"** This means that the frame erasure ratio was within the limits set with the commands explained in this section.

:CALCulate:GSM:RFRX:RBER:FER:LIMit:STATe

Syntax	:CALC:GSM:RFRX:RBER:FER:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFRX:RBER:FER measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFRX:RBER:FER:LIM:STAT ON** Switches the limit check of the frame erasure ratio measurement on.

:CALCulate:GSM :RFRX:RBER:FER:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM :RFRX:RBER:FER:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **2.5**.
Description	Sets the **upper** limit for the frame erasure ratio measurement. **Note:** For further details regarding the FER, please refer to section RBER live display.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:RBER:FER:LIM:UPP 4.5** Sets the upper limit for the FER measurement results to 4.5%.

:CALCulate:GSM :RFRX:RBER:FER:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFRX:RBER:FER:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**. The default value is **0.0**.
Description	Sets the **lower** limit for the frame erasure ratio measurement.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:RBER:FER:LIM:LOW 0** Sets the lower limit for the FER measurement results to 0%.

:CALCulate:GSM:RFRX:BER:ALL:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFRX:BER:ALL:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	This command delivers 3 boolean numbers in a string, separated by commas. These numbers indicate, which one of the (nonresidual) bit error ratio measurements did violate the corresponding test limits. While a **0** delivered back on any position means that the related test was passed without exceeding the limits, a **1** indicates that at least one of the limits was violated by the measurement result value. The position of the boolean number within the string returned indicates the result of the check for the following measurements: 1. the bit error ratio measurement of the class Ia bits, 2. the bit error ratio measurement of the class Ib bits and, 3. the bit error ratio measurement of the class II bits. Notes – For further details regarding bit classes, please refer to section voice coding. – For a description of the measurements performed, refer to the description of the `:MEAS:GSM:RFRX:BER:ALL` command.
Example	**`:MEAS:GSM:RFRX:BER:ALL 3`** **`:CALC:GSM:RFRX:BER:ALL:LIM?`** String returned:**`"0,0,1"`** The **1** on position 3 indicates that there was a violation of at least one limit of the bit error ratio measurement of the class II bits.

`:CALCulate:GSM:RFRX:BER:ALL:LIMit:STATe`

Syntax	`:CALCulate:GSM:RFRX:BER:ALL:LIMit:STATe <PredefExp>`
Parameters	**`PredefExp`** is one of the following predefined expressions: **`ON|OFF`**. Default is **`ON`**.
Description	This command switches the check of the measurement result values of the `MEAS:GSM:RFRX:BER:ALL` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**`:CALC:GSM:RFRX:BER:ALL:LIM:STAT OFF`** Switches the limit check of all (nonresidual) bit error ratio measurements off.

`:CALCulate:GSM`
`:RFRX:BER:ALL:LIMit:UPPer[:DATA]`

Syntax	`:CALCulate:GSM`
	`:RFRX:BER:ALL:LIMit:UPPer[:DATA]`
	`<real1>,<real2>,<real3>`

Parameters	**`realx`** are three floating point real numbers. The minimum value for all **`realx`** is **`0.0`**, the maximum value **`100.0`**. The minimum resolution possible for all **`realx`** is **`0.1`**. The default value for all **`realx`** is **`2.5`**.
Description	Sets the **upper** limits for the three (nonresidual) bit error ratio measurements with one command. The order of the floating point real numbers defines their meaning: **`real1`** represents the upper limit of the bit error ratio measurement for the class Ia bits, **`real2`** represents the upper limit of the bit error ratio measurement for the class Ib bits and, **`real3`** represents the upper limit of the bit error ratio measurement for the class II bits.
Query	The query form of this command is not supported.

Example	**:CALC:GSM:RFRX:BER:ALL:LIM:UPP 1,2,2.8** Sets the upper limit of the BER measurements to 1.0% for class Ia bits, 2.0% for class Ib bits and 2.8% for class II bits.

:CALCulate:GSM :RFRX:BER:ALL:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFRX:BER:ALL:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>
Parameters	**realx** are three floating point real numbers. The minimum value for all **realx** is **0.0**, the maximum value **100.0**. The minimum resolution possible for all **realx** is **0.1**. The default value for all **realx** is **0.0**.
Description	Sets the **lower** limits for the three (nonresidual) bit error ratio measurements with one command. The order of the floating point real numbers defines their meaning: **real1** represents the lower limit of the bit error ratio measurement for the class Ia bits, **real2** represents the lower limit of the bit error ratio measurement for the class Ib bits and, **real3** represents the lower limit of the bit error ratio measurement for the class II bits.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:BER:ALL:LIM:LOW** Sets the lower limit for all three BER measurements (class Ia, class Ib and class II) to 0%.

:CALCulate:GSM:RFRX:BER:CIA:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFRX:BER:CIA:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating, whether the result value of the (nonresidual) bit error ratio measurement of the **class Ia** bits did violate the user-definable test limits. While a **0** delivered back means that the related test was passed without exceeding the limits, a **1** indicates that at least one limit was violated by the measurement result value.
Example	**:MEAS:GSM:ARRay:RFRX:BER:CIA 5** **:CALC:GSM:RFRX:BER:CIA:LIM?** String returned:**"1"** This means that the bit error ratio measured for the class Ia bits was outside the limits set with the commands explained in this section.

:CALCulate:GSM:RFRX:BER:CIA:LIMit:STATe

Syntax	:CALCulate:GSM:RFRX:BER:CIA:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.

Description	This command switches the check of the measurement result values of the MEAS:GSM:RFRX:BER:CIA measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFRX:BER:CIA:LIM:STAT ON** Switches the limit check of the (nonresidual) bit error ratio measurement for the class Ia bits on.

:CALCulate:GSM :RFRX:BER:CIA:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM :RFRX:BER:CIA:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **2.5**.
Description	Sets the **upper** limit for the (nonresidual) bit error ratio measurement of the **class Ia** bits.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:BER:CIA:LIM:UPP 2** Sets the upper limit for bit error rate measurements on class Ia bits to 2.0%.

:CALCulate:GSM :RFRX:BER:CIA:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFRX:BER:CIA:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.
Description	Sets the **lower** limit for the (nonresidual) bit error ratio measurement of the **class Ia** bits.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:BER:CIA:LIM:LOW 0.0** Sets the lower limit for BER measurements on class Ia bits to 0.

:CALCulate:GSM:RFRX:BER:CIB:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFRX:BER:CIB:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating, whether the result value of the (nonresidual) bit error ratio measurement of the **class Ib** bits did violate the user-definable test limits. While a **0** delivered back means that the related test was passed without exceeding the limits, a **1** indicates that at least one limit was violated by the measurement result value.

Example	**:MEAS:GSM:RFRX:BER:CIB** **:CALC:GSM:RFRX:BER:CIB:LIM?** String returned:**"1"** This means that the bit error ratio measured for the class Ib bits was outside the limits set with the commands explained in this section.

:CALCulate:GSM:RFRX:BER:CIB:LIMit:STATe

Syntax	:CALCulate:GSM:RFRX:BER:CIB:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFRX:BER:CIB measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFRX:BER:CIB:LIM:STAT ON** Switches the limit check of the (nonresidual) bit error ratio measurement for the class Ib bits on.

:CALCulate:GSM :RFRX:BER:CIB:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM :RFRX:BER:CIB:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **2.5**.
Description	Sets the **upper** limit for the (nonresidual) bit error ratio measurement of the **class Ib** bits.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:BER:CIB:LIM:UPP 2.0** Sets the upper limit for class Ib bits to 2%.

:CALCulate:GSM :RFRX:BER:CIB:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFRX:BER:CIB:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.
Description	Sets the **lower** limit for the (nonresidual) bit error ratio measurement of the **class Ib** bits.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:BER:CIB:LIM:LOW 0** Sets the lower limit for class Ib bits to 0%.

:CALCulate:GSM:RFRX:BER:CII:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFRX:BER:CII:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating, whether the result value of the (nonresidual) bit error ratio measurement of the **class II** bits did violate the user-definable test limits. While a **0** delivered back means that the related test was passed without exceeding the limits, a **1** indicates that at least one limit was violated by the measurement result value.
Example	**:MEAS:GSM:ARRay:RFRX:BER:CII 3** **:CALC:GSM:RFRX:BER:CII:LIM?** String returned:**"1"** This means that the bit error ratio measured for the class II bits was outside the limits set with the commands explained in this section.

:CALCulate:GSM:RFRX:BER:CII:LIMit:STATe

Syntax	:CALCulate:GSM:RFRX:BER:CII:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFRX:BER:CII measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFRX:BER:CII:LIM:STAT ON** Switches the limit check of the (nonresidual) bit error ratio measurement for the class II bits on.

:CALCulate:GSM :RFRX:BER:CII:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GSM :RFRX:BER:CII:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **2.5**.
Description	Sets the **upper** limit for the (nonresidual) bit error ratio measurement of the **class II** bits.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:BER:CII:LIM:UPP 7.5** Sets the upper limit of the class II BER measurements to 7.5%.

:CALCulate:GSM :RFRX:BER:CII:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFRX:BER:CII:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.
Description	Sets the **lower** limit for the (nonresidual) bit error ratio measurement of the **class II** bits.
Query	The query form of this command is not supported.
Example	**:CALC:SGM:RFRX:BER:CII:LIM:LOW 0** Sets the lower limit of the class II BER measurements to 0%.

:CALCulate:GSM:RFRX:FBER:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFRX:FBER:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating, whether the result value of the fast bit error ratio measurement (C-type loop) did violate the user-definable test limits. While a **0** delivered back means that the related test was passed without exceeding the limits, a **1** indicates that at least one limit was violated by the measurement result value. **Note:** For further details regarding the fast BER measurement, please refer to section FBER live display.
Example	**:MEAS:GSM:RFRX:FBER** **:CALC:GSM:RFRX:FBER:LIM?** String returned:**"1"** This means that the fast bit error ratio measured was outside the limits set with the commands explained in this section.

:CALCulate:GSM:RFRX:FBER:LIMit:STATe

Syntax	:CALCulate:GSM:RFRX:FBER:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFRX:FBER measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFRX:FBER:LIM:STAT ON** Switches the limit check of the fast bit error ratio measurement on.

:CALCulate:GSM:RFRX:FBER:LIMit:UPPer[:DATA]

Syntax	:CALC:GSM:RFRX:FBER:LIM:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **2.5**.
Description	Sets the **upper** limit for the fast bit error ratio measurement result value.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:FBER:LIM:UPP 7.5** Sets the upper limit of the FER measurement results to 7.5%.

:CALCulate:GSM:RFRX:FBER:LIMit:LOWer[:DATA]

Syntax	:CALC:GSM:RFRX:FBER:LIM:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.
Description	Sets the **lower** limit for the fast bit error ratio measurement result value. **Note:** For further details regarding the fast BER measurement, please refer to section FBER live display.
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFRX:FBER:LIM:LOW 0.0** Sets the lower limit of the FER measurement results to 0%.

:CALCulate:GSM:RFSPectrum:ACPM:MAVerage?

Syntax	:CALCulate:GSM:RFSPectrum:ACPM:MAVerage?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Calculates and returns the **average** measurement result value of the ACPM measurement started last. The string delivered back will contain as many average values as measurement types performed. The single average values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:GSM:ARR:RFSP:ACPM:MOD** **:CALC:RFSP:ACPM:MAV?**

:CALCulate:GSM:RFSPectrum:ACPM:MMINimum?

Syntax	:CALCulate:GSM:RFSPectrum:ACPM:MMINimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	Returns the **minimum** measurement result value of the ACPM measurement started last. The string delivered back will contain as many minimum values as measurement types performed. The single minimum values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:GSM:ARR:RFSP:ACPM:MOD** **:CALCulate:RFSP:ACPM:MMIN?**

:CALCulate:GSM:RFSPectrum:ACPM:MMAXimum?

Syntax	:CALCulate:GSM:RFSPectrum:ACPM:MMAXimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **maximum** measurement result value of the ACPM measurement started last. The string delivered back will contain as many maximum values as measurement types performed. The single maximum values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:GSM:ARR:RFSP:ACPM:MOD** **:CALC:RFSP:ACPM:MMAX?**

:CALCulate:GSM:RFSPectrum:ACPM:MSIGma?

Syntax	:CALCulate:GSM:RFSPectrum:ACPM:MSIGma?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **mean value** and the **standard deviation** of the RF spectrum measurement. In the case of the ACP Mod measurement, the string delivered back contains 27 pairs of mean values and standard deviation values for the ACP Mod measurement. In the ACP Transient case, these are 23 pairs of mean values and standard deviation values. In either case, the values are floating point real values and separated by commas.
Example	**:CALC:RES** **:MEAS:GSM:ARR:RFSP:ACPM:MOD** **...(other SCPI commands)...** **:CALCulate:GSM:RFSP:ACPM:MSIG?**

:CALCulate:GSM:RFSPectrum
:ACPM:MODulation:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFSPectrum :ACPM:MODulation:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query This command delivers 23 boolean numbers in a string, separated by commas. These numbers indicate whether the 23 single measurement result values of the MEAS:GSM:RFSP:ACPM:MOD measurement did violate their individual limits (set with the commands explained below). Every boolean number represents the result of the limit check of one of the 23 measurement result values. The first value will give the result of the limit check at -1.8 MHz offset to the carrier, the second one for -1.6 MHz offset to the carrier and so on.
While a **0** delivered back on any position means that the related measurement result value did not exceed the limits, a **1** indicates that at least one limit was violated by the measurement result value.
Note: According to GSM specification 05.05, the **MEAS:GSM:RFSP:ACPM:MOD** measurement will be performed on (±)100 kHz and (±)250 kHz offset from the carrier as well as on all multiples of (±)200 kHz offset from the carrier up to (and including) an offset of (±)1.8 MHz. If one includes the carrier frequency as well (0.0 dBc), this adds up to the 23 measurement values returned. All measurements will be performed with a measuring bandwidth of 30 kHz.

Example
```
:MEAS:GSM:BLOC:MSP:AVG 20
:CALC:GSM:RFSP:ACPM:MOD:LIM?
```
String returned:
```
"0,0,0,0,0,0,0,0,1, 0,0,0,0,0,0,0,0,0,0,0,0,0"
```
The **1** on position 9 indicates that there was a violation of at least one limit by the measurement result value taken at an offset of -0.25 MHz from the carrier.

:CALCulate:GSM:RFSPectrum:ACPM:MODulation:LIMit :STATe

Syntax :CALC:GSM:RFSP:ACPM:MOD:LIMit:STATe <PredefExp>

Parameters **PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.

Description This command switches the check of the measurement result values of the MEAS:GSM:RFSP:ACPM:MOD measurement against their user-definable limits either on or off.

Query There is no query form of this command available.

Example
```
:CALC:GSM:RFSP:ACPM:MOD:LIM:STAT ON
```
Switches the limit check of the ACPM of the modulation spectrum measurement on.

:CALCulate:GSM :RFSPectrum:ACPM:MODulation:LIMit:UPPer[:DATA]

Syntax
```
:CALCulate:GSM
:RFSPectrum:ACPM:MODulation:LIMit:UPPer[:DATA]
<real1>,<real2>,<real3>,<real4>,
<real5>,<real6>,<real7>,<real8>,
<real9>,<real10>,<real11>,<real12>,
<real13>,<real14>,<real15>,<real16>,
<real17>,<real18>,<real19>,<real20>,
<real21>,<real22>,<real23>
```

Parameters	**realx** are 23 floating point real numbers. The minimum value for all **realx** is **-100.0**, the maximum value **10.0**. The minimum resolution possible for all **realx** is **0.01**. The default values are: **-60.0** for **real1**...**real8**, **-33.0**> for **real9**, **-30.0** for **real10**, **0.5** for **real11**, **0.5** for **real12**, **0.5** for **real13**, **-30.0** for **real14**, **-33.0** for **real15** and **-60.0** for **real16**...**real23**.
Description	Sets the **upper** limits for the 23 single measurement result values of the ACPM modulation spectrum measurement. The single data values are interpreted as dBc (dB relative to the power level measured at the transmission frequency).
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFSP:ACPM:MOD:LIM:UPP -65,-60,-60,-60,-60,-60,-60,-60,-33,-30,0.5,0.5,0.5,-30,-33,-60,-60,-60,-60,-60,-60,-60,-65** Sets the upper limits of the ACP modulation spectrum to these values. The first value applies to -1.6 MHz, the twelfth value to the carrier frequency and the 23rd value to a +1.6 MHz offset.

:CALCulate:GSM :RFSPectrum:ACPM:MODulation:LIMit:LOWer[:DATA]

Syntax	:CALC:GSM :RFSPectrum:ACPM:MODulation:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>, <real9>,<real10>,<real11>,<real12>, <real13>,<real14>,<real15>,<real16>, <real17>,<real18>,<real19>,<real20>, <real21>,<real22>,<real23>
Parameters	**realx** are 23 floating point real numbers. The minimum value for all **realx** is **-100.0**, the maximum value **10.0**. The minimum resolution possible for all **realx** is **0.01**. The default value for all **realx** is **-100.0**.
Description	Sets the **lower** limits for the 23 single measurement result values of the ACPM modulation spectrum measurement. The single data values are interpreted as dBc (dB relative to the power level measured at the transmission frequency).
Query	The query form of this command is not supported.
Example	**:CALC:GSM:RFSP:ACPM:MOD:LIM:LOW -100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,-100,** Sets the lower limits of the ACP modulation measurement to zero for all 23 measured frequencies.

:CALCulate:GSM:RFSPectrum :ACPM:TRANsient:LIMit[:FAIL]?

Syntax	:CALCulate:GSM:RFSPectrum :ACPM:TRANsient:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers 27 boolean numbers in a string, separated by commas. These numbers indicate whether the 27 measurement result values of the MEAS:GSM:RFSP:ACPM:TRAN measurement did violate their individual limits (set with the commands explained below). Every boolean number represents the result of the limit check for one of the 27 measurement result values. The first value will give the result of the limit check at -1.83 MHz offset to the carrier, the second one for -1.80 MHz offset to the carrier and so on. While a **0** delivered back on any position means that the related measurement result value did not exceed the limits, a **1** indicates that at least one limit was violated by the measurement result value. **Note:** According to GSM specification 05.05, this measurement will be performed on 0, (±)400, (±)600, (±)1.200 and (±)1.800 kHz offset from the carrier. All measurements will be performed with a measuring bandwidth of 30 kHz. For additional information, also the measurement values of the 30 kHz windows below and above the frequencies specified by GSM standard 05.05 will be taken into account. This adds up to 9 x 3 = 27 measurement result values.
Example	**:MEAS:GSM:ARRay:RFSP:ACPM:TRANsient 10** **:CALC:GSM:RFSP:ACPM:TRAN:LIM?** String returned:**"0,0,0,0,0,0,0,0,1,** **0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0"** The **1** on position 9 indicates that there was a violation of at least one limit by the measurement result value taken at an offset of -0.57 MHz from the carrier.

:CALCulate:GSM:RFSPectrum:ACPM:TRANsient:LIMit :STATe

Syntax	:CALCulate:GSM:RFSPectrum:ACPM:TRANsient:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:GSM:RFSP:ACPM:TRAN measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFSP:ACPM:TRAN:LIM:STAT ON** Switches the limit check of the ACPM of the spectrum due to switching transients on.

:CALCulate:GSM :RFSPectrum:ACPM:TRANsient:LIMit:UPPer[:DATA]

Syntax
```
:CALCulate:GSM
:RFSPectrum:ACPM:TRANsient:LIMit:UPPer[:DATA]
<real1>,<real2>,<real3>,<real4>,
<real5>,<real6>,<real7>,<real8>,
<real9>,<real10>,<real11>,<real12>,
<real13>,<real14>,<real15>,<real16>,
<real17>,<real18>,<real19>,<real20>,
<real21>,<real22>,<real23>,<real24>,
<real25>,<real26>,<real27>
```

Parameters
realx are 27 floating point real numbers.
The minimum value for all **realx** is **-100.0**, the maximum value **10.0**. The minimum resolution possible for all **realx** is **0.01**. The default value for all **realx** is **0.0**.

Description
Sets the **upper** limits for the 27 single measurement result values of the ACPM measurement of the spectrum due to switching transients. The physical dimension of the single data values is dBm.

Query
The query form of this command delivers the current settings of the upper limits. The string returned will contain 27 floating point real numbers, separated by commas, in the order and meaning as explained above.

Example
:CALC:GSM:RFSP:ACPM:TRAN:LIM:UPP?
This command will return the 27 upper limit values.

:CALCulate:GSM :RFSPectrum:ACPM:TRANsient:LIMit:LOWer[:DATA]

Syntax
```
:CALC:GSM
:RFSP:ACPM:TRAN:LIMit:LOWer[:DATA]
<real1>,<real2>,<real3>,<real4>,
<real5>,<real6>,<real7>,<real8>,
<real9>,<real10>,<real11>,<real12>,
<real13>,<real14>,<real15>,<real16>,
<real17>,<real18>,<real19>,<real20>,
<real21>,<real22>,<real23>,<real24>,
<real25>,<real26>,<real27>
```

Parameters
realx are 27 floating point real numbers.
The minimum value for all **realx** is **-100.0**, the maximum value **10.0**. The minimum resolution possible for all **realx** is **0.01**. The default value for all **realx** is **0.0**.

Description
Sets the **lower** limits for the 27 single measurement result values of the ACPM measurement of the spectrum due to switching transients. The physical dimension of the single data values is dBm.

Query
The query form of this command delivers the current settings of the lower limits. The string returned will contain 27 floating point real numbers, separated by commas, in the order and meaning as explained above.

Example
:CALC:GSM:RFSP:ACPM:TRAN:LIM:LOW?
This command will return the 27 lower limit values.

:CALCulate:GSM:RFSPectrum:MSPectrum:VALue

Syntax	:CALCulate:GSM:RFSPectrum:MSPectrum:VALue <real1>
Parameters	**real1** is a floating point real number. The minimum value is **-1,000,000.0**, the maximum is **1,000,000.0**. The minimum resolution possible is **0.1**, the default value is **-200.0**.
Description	There is only a query form of this command available.
Query	The query form of this command delivers an (interpolated) measurement result value of the modulation spectrum curve at the frequency specified with the **real1** parameter. **real1** is interpreted as an offset from the carrier (i.e. the transmission frequency). The string returned will contain one floating point real number. The physical dimension of the measurement result value returned is dBc. Notes – Please note that you need to have a spectrum measurement taken before this command will return any meaningful result. We recommend to use the :MEAS:BLOC:MSP:AVG command to do so. – The **:CALC:GSM:RFSP:MSP:VAL** command is not really a measurement - it is an interpolation of the spectrum curve for the frequency specified.
Example	**:MEAS:BLOC:MSP:AVG 20** **:CALC:GPRS:RFSP:MSP:VAL? -50000** This command will interpolate a modulation spectrum measurement at an offset of 50 kHz to the carrier. The value returned in this example is: **"-4.7"**.

:CALCulate:GSM:RFSPectrum:MSPectrum:MAXPeak

Syntax	:CALCulate:GSM:RFSPectrum:MSPectrum:MAXPeak?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	The query form of this command delivers back the frequency offset (in Hz) from the carrier and the maximum value of the measured modulation spectrum curve. The string returned will contain two floating point real numbers. The physical dimension of the measurement result values returned is for the first value Hz and for the second value dBc. Notes Please note that you need to have a spectrum measurement taken before this command will return any meaningful result. We recommend to use the :MEAS:BLOC:MSP:AVG command to do so. The :CALC:GSM:RFSP:MSP:MAXP command is not really a measurement - it is an readout of the spectrum curve.
Example	:MEAS:GSM:BLOC:MSP:AVG 20 :CALC:GSM:RFSP:MSP:MAXP? This command will read at the modulation spectrum measurement maximum the frequency offset to the carrier and the measurement result value. The values returned in this example is: "-67700.0,-5.2".

:CALCulate:AFANalyser:MAVerage?

Syntax	:CALCulate:AFANalyser:MAVerage?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Calculates and returns the **average** measurement result value of the Audio measurement started last. The string delivered back will contain as many average values as measurement types performed. The single average values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:ARR:AFAN:ALL** **:CALC:AFAN:MAV?**

:CALCulate:AFANalyser:MMINimum?

Syntax	:CALCulate:AFANalyser:MMINimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **minimum** measurement result value of the Audio measurement started last. The string delivered back will contain as many minimum values as measurement types performed. The single minimum values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:ARR:AFAN:ALL** **:CALC:AFAN:MMIN?**

:CALCulate:AFANalyser:MMAXimum?

Syntax	:CALCulate:AFANalyser:MMAXimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **maximum** measurement result value of the Audio measurement started last. The string delivered back will contain as many maximum values as measurement types performed. The single maximum values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:ARR:AFAN:ALL** **:CALC:AFAN:MMAX?**

:CALCulate:AFANalyser:MSIGma?

Syntax	:CALCulate:AFANalyser:MSIGma?
Parameters	There are no parameters.

Description	There is solely a query form of this command available.
Query	Returns the **mean value** and the **standard deviation** of the audio measurement results since the last measurement start. The string delivered back contains as many pairs of mean and standard deviation values as measurement types performed. The individual values all have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:AFAN:ACV** **...(other SCPI commands)...** **:CALC:AFAN:MSIG?** In this example, first a continuous measurement of the audio voltage is started. After some time, the **:CALC:AFAN:MSIGma** command is used to read out the mean voltage and its standard deviation. The string returned in this example is **"1.1, 0.2"**.

:CALC:AFANalyser:ALL:LIMit[:FAIL]?

Syntax	:CALC:AFANalyser:ALL:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers six boolean numbers in a string, separated by commas. These numbers indicate whether the six single measurement result values of the MEAS:AFAN:ALL measurement did violate their individual limits (set with the commands explained below) or not. The position of the boolean number within the string returned indicates the result of the check of following measurements: 1. the peak-to-peak measurement of the AC voltage, 2. the root-mean square value of the AC voltage, 3. the RMS-valued AC ripple on a DC voltage, 4. the audio frequency measurement, 5. the measurement of the third-harmonic distortion (THD) of a sine wave and 6. the SINAD measurement. While a **0** on any position indicates that the related measurement result value is within the limits specified, a **1** indicates that the measurement result value did violate at least one of the limits set for the related test.
Example	**:MEAS:AFAN:ALL** **:CALC:AFAN:ALL:LIM?** String returned:**"0,0,0,0,1,0"** The **1** on position 5 indicates that the distortion measurement is off its limits.

:CALCulate:AFANalyser:ALL:LIMit:STATe

Syntax	:CALCulate:AFANalyser:ALL:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:AFAN:ALL measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.

Example	**:CALC:AFAN:ALL:LIM:STAT ON** Switches the limit check of the main audio measurements on.

:CALCulate:AFANalyser:ALL:LIMit:UPPer[:DATA]

Syntax	:CALCulate:AFANalyser:ALL:LIMit:UPPer[:DATA] <real1>,<real2>,<real3>,<real4>, <real5>,<real6>
Parameters	**realx** are six floating point real numbers. **real1**: The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.0001**, the default value is **5.0**. **real2**: The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.0001**, the default value is **5.0**. **real3**: The minimum value is **–40.0**, the maximum is **40.0**. The minimum resolution possible is **0.0001**, the default value is **5.0**. **real4**: The minimum value is **0.0**, the maximum is **20,000.0**. The minimum resolution possible is **1.0**, the default value is **5,000.0**. **real5**: The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **5.0**. **real6**: The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **20.0**.
Description	Sets the **upper** limits for the six main audio frequency measurements with one command. The order of the floating point real numbers defines their meaning: **real1** represents the upper limit of the peak-to-peak AC voltage measurement; the physical dimension of the number stated is V(pp). **real2** represents the upper limit of the RMS-valued AC voltage measurement; the physical dimension of the number stated is V(rms). **real3** represents the upper limit of the RMS-valued measurement of an AC voltage ripple on a DC voltage; the physical dimension of the number stated is V(rms). **real4** represents the upper limit of the audio frequency measurement; the physical dimension of the number stated is Hertz. **real5** represents the upper limit of the distortion measurement, carried out on the third harmonic of a sine wave. The physical dimension of the number stated is percentage. **real6** represents the upper limit of the SINAD signal-to-noise ratio measurement; the physical dimension of the number stated is dB.
Query	The query form of this command is not supported.
Example	**:CALC:AFAN:ALL:LIM:UPP 3,3,1,4000,5,100** Sets the upper limits for the audio measurements: 3 V for the peak-to-peak AC voltage, 3 V for the rms-valued AC voltage, 1 V for the rms-valued ripple voltage, 4 kHz for the audio frequency, 5% for THD, and 100 dB for SINAD.

:CALCulate:AFANalyser:ALL:LIMit:LOWer[:DATA]

Syntax
:CALCulate:AFANalyser:ALL:LIMit:LOWer[:DATA]
<real1>,<real2>,<real3>,<real4>, <real5>,<real6>

Parameters
realx are six floating point real numbers.
real1: The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.0001**, the default value is **1.0**.
real2: The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.0001**, the default value is **1.0**.
real3: The minimum value is **-40.0**, the maximum is **40.0**. The minimum resolution possible is **0.0001**, the default value is **-5.0**.
real4: The minimum value is **0.0**, the maximum is **20,000.0**. The minimum resolution possible is **1.0**, the default value is **1,000.0**.
real5: The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.
real6: The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.

Description
Sets the **lower** limits for the six main audio frequency measurements with one command. The order of the floating point real numbers defines their meaning:
real1 represents the lower limit of the peak-to-peak AC voltage measurement; the physical dimension of the number stated is V(pp).
real2 represents the lower limit of the RMS-valued AC voltage measurement; the physical dimension of the number stated is V(rms).
real3 represents the lower limit of the RMS-valued measurement of an AC voltage ripple on a DC voltage; the physical dimension of the number stated is V(rms).
real4 represents the lower limit of the audio frequency measurement; the physical dimension of the number stated is Hertz.
real5 represents the lower limit of the distortion measurement, carried out on the third harmonic of a sine wave. The physical dimension of the number stated is percentage.
real6 represents the lower limit of the SINAD signal-to-noise ratio measurement; the physical dimension of the number stated is dB.

Query
The query form of this command is not supported.

Example
:CALC:AFAN:ALL:LIM:LOW 1,1,0,400,0,25
Sets the lower limits for the audio measurements.

:CALCulate:AFANalyser :ACVoltage:PPEAk:LIMit[:FAIL]?

Syntax
:CALCulate:AFANalyser
:ACVoltage:PPEAk:LIMit[:FAIL]?

Parameters
There are no parameters.

Description
There is solely a query form of this command available.

Query
This command delivers one boolean number, indicating whether the user-definable limits of the peak-to-peak measurement of the AC voltage applied to the audio analyzer were violated by a measurement result value or not.
A **0** means that the measurement result value is within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.

Example	**:MEAS:ARRay:AFAN:ACV:PPEA 10** **:CALC:AFAN:ACV:PPEA:LIM?** String returned: **"1"** This string delivered back indicates that at least one measurement result value did violate the limits of the peak-to-peak AC measurement.

:CALCulate:AFANalyser:ACVoltage:PEAKp:LIMit:STATe

Syntax	:CALCulate:AFANalyser:ACVoltage:PEAKp:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:AFAN:ACV:PEAK measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:AFAN:ACV:PEAK:LIM:STAT ON** Switches the limit check of the peak-to-peak AC voltage measurement on.

:CALCulate:AFANalyser :ACVoltage:PEAKp:LIMit:UPPer[:DATA]

Syntax	:CALCulate:AFANalyser :ACVoltage:PEAKp:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.0001**, the default value is **5.0**.
Description	Sets the **upper** limit for the peak-to-peak measurement of the AC voltage applied to the audio analyzer. The physical dimension of the number stated is V(pp).
Query	The query form of this command is not available.
Example	**:CALC:AFAN:ACV:PEAK:LIM:UPP 3** Sets the upper limit of the peak-to-peak AC voltage measurement to 3 V.

:CALCulate:AFANanalyser :ACVoltage:PEAKp:LIMit:LOWer[:DATA]

Syntax	:CALCulate:AFANanalyser :ACVoltage:PEAKp:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.0001**, the default value is **1.0**.
Description	Sets the **lower** limit for the peak-to-peak measurement of the AC voltage applied to the audio analyzer. The physical dimension of the number stated is V(pp).
Query	The query form of this command is not available.
Example	**:CALC:AFAN:ACV:PEAK:LIM:LOW 0** Sets the lower limit of the peak-to-peak AC voltage measurement to 0 V.

:CALCulate:AFANalyser :ACVoltage:RMS:LIMit[:FAIL]?

Syntax	:CALCulate:AFANalyser :ACVoltage:RMS:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the RMS-valued measurement of the AC voltage applied to the audio analyzer were violated by a measurement result value or not. A **0** means that the measurement result value is within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:MEAS:ARRay:AFAN:ACV:RMS 5** **:CALC:AFAN:ACV:RMS:LIM?** String returned: **"0"** This string delivered back indicates that there was no violation of the limits set.

:CALCulate:AFANalyser :ACVoltage:RMS:LIMit:STATe

Syntax	:CALCulate:AFANalyser :ACVoltage:RMS:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:AFAN:ACV:RMS measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:AFAN:ACV:RMS:LIM:STAT ON** Switches the limit check of the RMS-valued AC voltage measurement on.

:CALCulate:AFANalyser :ACVoltage:RMS:LIMit:UPPer[:DATA]

Syntax	:CALCulate:AFANalyser :ACVoltage:RMS:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.0001**, the default value is **5.0**.
Description	Sets the **upper** limit for the RMS-valued measurement of the AC voltage applied to the audio analyzer. The physical dimension of the number stated is V(rms).
Query	The query form of this command is not available.
Example	**:CALC:AFAN:ACV:RMS:LIM:UPP 3** Sets the upper limit of the rms-valued AC voltage measurement to 3 V.

:CALCulate:AFANalyser :ACVoltage:RMS:LIMit:LOWer[:DATA]

Syntax	:CALCulate:AFANalyser :ACVoltage:RMS:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **30.0**. The minimum resolution possible is **0.0001**, the default value is **1.0**.
Description	Sets the **lower** limit for the RMS-valued measurement of the AC voltage applied to the audio analyzer. The physical dimension of the number stated is V(rms).
Query	The query form of this command is not available.
Example	**:CALC:AFAN:ACV:RMS:LIM:LOW 0** Sets the lower limit of the rms-valued AC voltage measurement to 0 V.

:CALCulate:AFANalyser :DCVoltage:LIMit[:FAIL]?

Syntax	:CALCulate:AFANalyser :DCVoltage:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the RMS-valued measurement of the AC ripple on a DC voltage were violated by a measurement result value or not. A **0** means that the measurement result value is within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:MEAS:ARRay:AFAN:DCV 10** **:CALC:AFAN:DCV:LIM?** String returned: **"0"** This string delivered back indicates that there has been no violation of the limits set.

:CALCulate:AFANalyser :DCVoltage:LIMit:STATe

Syntax	:CALCulate:AFANalyser :DCVoltage:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the :MEAS:AFAN:DCV measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:AFAN:DCV:LIM:STAT ON** Switches the limit check of the RMS-valued AC voltage ripple measurement of a DC voltage on.

:CALCulate:AFANalyser :DCVoltage:LIMit:UPPer[:DATA]

Syntax	:CALCulate:AFANalyser :DCVoltage:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **-40.0**, the maximum is **40.0**. The minimum resolution possible is **0.0001**, the default value is **5.0**.
Description	Sets the **upper** limit for the RMS-valued measurement of the AC voltage component on a DC voltage, applied to the audio analyzer. The physical dimension of the number stated is V(rms).
Query	The query form of this command is not supported.
Example	**:CALC:AFAN:DCV:LIM:UPP 6** Sets the upper limit of the rms-valued ripple voltage measurement to 6 V.

:CALCulate:AFANanalyser :DCVoltage:LIMit:LOWer[:DATA]

Syntax	:CALCulate:AFANanalyser :DCVoltage:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **-40.0**, the maximum is **40.0**. The minimum resolution possible is **0.0001**, the default value is **-5.0**.
Description	Sets the **lower** limit for the RMS-valued measurement of the AC voltage component on a DC voltage, applied to the audio analyzer. The physical dimension of the number stated is V(rms).
Query	The query form of this command is not supported.
Example	**:CALC:AFAN:DCV:LIM:LOW 1.75** Sets the upper limit of the rms-valued ripple voltage measurement to 1.75 V.

:CALCulate:AFANalyser :FREQuency:LIMit[:FAIL]?

Syntax	:CALCulate:AFANalyser :FREQuency:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the AF frequency measurement were violated by a measurement result value or not. A **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:MEAS:AFAN:FREQ** **:CALC:AFAN:FREQ:LIM?** String returned:**"1"** This string delivered back indicates that there has been a violation of the limits set.

:CALCulate:AFANalyser:FREQuency:LIMit:STAte?

Syntax	:CALCulate:AFANalyser:FREQuency:LIMit:STAte? <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:AFAN:FREQ measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:AFAN:FREQ:LIM:STAT ON** Switches the limit check of the AF frequency measurement on.

:CALCulate:AFANalyser :FREQuency:LIMit:UPPer[:DATA]

Syntax	:CALCulate:AFANalyser :FREQuency:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **20,000.0**. The minimum resolution possible is **1.0**, the default value is **5,000.0**.
Description	Sets the **upper** limit for the AF frequency measurement. The physical dimension of the number stated is Hertz.
Query	The query form of this command is not available.
Example	**:CALC:AFAN:FREQ:LIM:UPP 10000** Sets the upper limit of the audio frequency measurement to 10 kHz.

:CALCulate:AFANalyser :FREQuency:LIMit:LOWer[:DATA]

Syntax	:CALCulate:AFANalyser :FREQuency:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **20,000.0**. The minimum resolution possible is **1.0**, the default value is **1,000.0**.
Description	Sets the **lower** limit for the AF frequency measurement. The physical dimension of the number stated is Hertz.
Query	The query form of this command is not available.
Example	**:CALC:AFAN:FREQ:LIM:LOW 950** Sets the lower limit of the audio frequency measurement to 950 Hz.

:CALCulate:AFANalyser :DISTortion:LIMit[:FAIL]?

Syntax	:CALCulate:AFANalyser :DISTortion:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the distortion measurement were violated by a measurement result value or not. A **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below. **Notes** – The distortion measurement is carried through on the third harmonic of a 1 kHz sine wave. – All audio measurements require the Audio Option to be installed on your Willtek 3100.
Example	**:MEAS:ARRay:AFAN:DIST 10** **:CALC:AFAN:DIST:LIM?** String returned: **"1"** This string delivered back indicates that there was a violation of the limits set.

:CALCulate:AFANalyser :DISTortion:AFAN:DIST:LIMit:STATe

Syntax	:CALCulate:AFANalyser :DISTortion:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:AFAN:DIST measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:AFAN:DIST:LIM:STAT ON** Switches the limit check of the distortion measurement on.

:CALCulate:AFANalyser :DISTortion:LIMit:UPPer[:DATA]

Syntax	:CALCulate:AFANalyser :DISTortion:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **5.0**.

Description	Sets the **upper** limit for the distortion measurement. The physical dimension of the number stated is percent (the relation of the AC voltage measured for the third harmonic of a 1 kHz sine wave in relation to the AC voltage measured for the nominal frequency).
Query	The query form of this command is not available.
Example	**:CALC:AFAN:DIST:LIM:UPP 4.5** Sets the upper limit of the distortion measurement to 4.5%.

:CALCulate:AFANalyser :DISTortion:LIMit:LOWer[:DATA]

Syntax	:CALCulate:AFANalyser :DISTortion:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.
Description	Sets the **lower** limit for the distortion measurement. The physical dimension of the number stated is percent (the relation of the AC voltage measured for the third harmonic of a 1 kHz sine wave in relation to the AC voltage measured for the nominal frequency).
Query	The query form of this command is not available.
Example	**:CALC:AFAN:DIST:LIM:LOW 0.0** Sets the lower limit of the distortion measurement to 0%.

:CALCulate:AFANalyser :SINad:LIMit[:FAIL]?

Syntax	:CALCulate:AFANalyser :SINad:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the SINAD measurement were violated by a measurement result value or not. A **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below. **Notes** – The SINAD measurement is carried through with a 1 kHz sine wave. – All audio measurements require the Audio Option to be installed on your Willtek 3100.
Example	**:MEAS:ARRay:AFAN:SIN 5** **:CALC:AFAN:SIN:LIM?** String returned: **"1"** This string delivered back indicates that there was a violation of the limits set.

:CALCulate:AFANalyser :SINad:LIMit:STATe

Syntax	:CALCulate:AFANalyser :SINad:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:AFAN:SIN measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:AFAN:SIN:LIM:STAT ON** Switches the limit check of the SINAD measurement on.

:CALCulate:AFANalyser :SINad:LIMit:UPPer[:DATA]

Syntax	:CALCulate:AFANalyser :SINad:LIMit:UPPer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **20.0**.
Description	Sets the **upper** limit for the SINAD measurement. The physical dimension of the number stated is dB.
Query	The query form of this command is not available.
Example	**:CALC:AFAN:SIN:LIM:UPP 100** Sets the upper limit of the SINAD measurement to 100%nbsp;dB.

:CALCulate:AFANalyser :SINad:LIMit:LOWer[:DATA]

Syntax	:CALCulate:AFANalyser :SINad:LIMit:LOWer[:DATA] <real1>
Parameters	**real1** is a floating point real number. The minimum value is **0.0**, the maximum is **100.0**. The minimum resolution possible is **0.1**, the default value is **0.0**.
Description	Sets the **lower** limit for the SINAD measurement. The physical dimension of the number stated is dB.
Query	The query form of this command is not available.
Example	**:CALC:AFAN:SIN:LIM:LOW 30** Sets the lower limit of the SINAD measurement to 30 dB.

:CALCulate:AFSPectrum:TEMPlate:LIMit[:FAIL]

Syntax	:CALCulate:AFSPectrum:TEMPlate:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	This command delivers ten boolean numbers, indicating whether the user-definable limits of the audio spectrum measurement were violated by a measurement result value or not. A **0** means that the measurement result values for the respective section of the template were within the limits set, while a **1** indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:MEAS:ARRay:AFSP 10** **:CALC:AFSP:TEMP:LIM:FAIL?** String returned: **"0,1,0,0,0,0,0,0,0,0"** This string delivered back indicates that there was a violation of the limits set for the second area.

:CALCulate:AFSPectrum:TEMPlate:LIMit:STATe

Syntax	:CALCulate:AFSPectrum:TEMPlate:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:AFSP:TEMPlate and MEAS:ARR:AFSP:TEMPlate measurements against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:GSM:RFTX:TEMP:LIM:STAT ON** Switches on the limit check of the audio spectrum measurement against the user-defined template.

:CALCulate:AFSPectrum:TEMPlate:LIMit:LOAD

Syntax	:CALCulate:AFSPectrum:TEMPlate:LIMit:LOAD <string1>
Parameters	**string1** is a string giving the name of the file containing the audio limits. The maximum length of **string1** is 50 characters. The default for **string1** is **"example.lmt"**.
Description	This command loads the audio template description file. Please note that the data contained in the file need to be activated (using the **CALC:AFSP:TEMP:LIM:STAT ON** command described above) before the template will have any effect on the measurement results.
Query	There is no query form of this command available.
Example	**:CALC:AFSP:LIM:LOAD "example.lmt"** Loads the audio spectrum template file example.lmt which resides in the /rapid/audiolmt directory.

:CALCulate:AFSPectrum:TEMPlate:LIMit:STORe

Syntax	:CALCulate:AFSPectrum:TEMPlate:LIMit:STORe <string1>
Parameters	**string1** is a string giving the name of the file in which the current audio limits are to be saved. The maximum length of **string1** is 50 characters.

Description	This command saves the current audio template in a file. The template can be defined by the FREQ, UPPer, LOWer and COMMent comments.
Query	There is no query form of this command available.
Example	**:CALC:AFSP:LIM:STORe "example.lmt"** Stores the currently defined audio spectrum template in the file example.lmt in the /rapid/audiolmt directory. If a file of this name already exists, it will be overwritten.

:CALCulate:AFSPectrum:TEMPlate:LIMit:COMMent

Syntax	:CALCulate:AFSPectrum:TEMPlate:LIMit:COMMent <string1>
Parameters	**string1** is a comment line related to the coupling loss data. The maximum length of **string1** is 80 characters.
Description	Defines a comment line for storage in the audio template file and for display in the **Comment line** field on the Limits screen. **Note:** The comment can be saved with the STORe command, together with the other template parameters.
Query	There is no query form of this command available.
Example	:CALC:AFSP:LIM:COMM "Audio template for Siemens S35"

:CALCulate:AFSPectrum:TEMPlate:LIMit:FREQuency

Syntax	:CALCulate:AFSPectrum:TEMPlate:LIMit:FREQuency <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>, <real9>,<real10>
Parameters	**real1** through **realxp** are floating point real numbers. The minimum value for all numbers is **0.0**, the maximum value **20000.0**. The minimum resolution possible for all **realxt** is **0.1**. The default values are: for **real0t**: **-41.0**, for **real1t**: **-28.0**, for **real2t**: **-18.0**, for **real3t**: **-10.0** for **real4t**: **0.0**, for **real5t**: **553.0** for **real6t**: **561.0**, for **real7t**: **571.0** for **real8t**: **580.0**. The minimum value for all **realxp** is **-150.0**, the maximum value **5.0**. The minimum resolution possible for all **realx** is **0.1**.
Description	Sets the frequency values for the user-definable audio spectrum template. The values are in Hz. They are used in conjunction with the defintions for the upper and lower limits.
Query	The query form of this command is not available.
Example	**:CALC:AFSP:TEMP:LIM:FREQ 25, 50, 100, 200, 500, 1000, 2000, 4000, 8000, 16000** This command sets the frequency values for the upper and lower limits. **Note:** These are the default values after an instrument reset.

:CALCulate:AFSPectrum:TEMPlate:LIMit:UPPer

Syntax	:CALCulate:AFSPectrum:TEMPlate:LIMit:UPPer <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>, <real9>,<real10>
Parameters	The **realx** values are floating point real numbers. The minimum value is **-120**, the maximum value is **+10**. The resolution for all values is **0.1**. The values are the upper limits for the audio template in dB, relative to the reference level.
Description	Sets the **upper** limits for the user-definable audio spectrum template. These upper limits refer to the frequencies defined in the **FREQuency** command above.
Query	The query form of this command is not available.
Example	**:CALC:AFSP:TEMP:LIM:UPP 5, 5, 5, 5, 5, 5, 5, 5, 5, 5** This command sets all upper limits to +5 dB at the frequencies defined with the **:CALC:AFSP:TEMP:LIM:FREQ** command.

:CALCulate:AFSPectrum:TEMPlate:LIMit:LOWer

Syntax	:CALCulate:AFSPectrum:TEMPlate:LIMit:LOWer <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>, <real9>,<real10>
Parameters	The **realx** values are floating point real numbers. The minimum value is **-120**, the maximum value is **+10**. The resolution for all values is **0.1**. The values are the lower limits for the audio template in dB, relative to the reference level.
Description	Sets the **lower** limits for the user-definable audio spectrum template. These upper limits refer to the frequencies defined in the **FREQuency** command above.
Query	The query form of this command is not available.
Example	**:CALC:AFSP:TEMP:LIM:LOW -25, -25, -25, -25, -25, -25, -25, -25, -25, -25** This command sets all lower limits to -25 dB at the frequencies defined with the **:CALC:AFSP:TEMP:LIM:FREQ** command.

:CALCulate:AFSPectrum:VALue?

Syntax	:CALCulate:AFSPectrum:VALue? <real1>
Parameters	**real1** is a floating point real number. It provides the position on the frequency axis for which the audio spectrum value is being requested.
Query	Provides the measurement value (audio spectrum level) for the given frequency value. If there is no measurement value at exactly this frequency, the value is interpolated.
Example	**:MEAS:BLOC:AFSP:AVG** **:CALC:AFSP:VAL? 1200** Delivers the audio spectrum value at the frequency of 1.2 kHz.

:CALCulate:AFSPectrum:MAXPeak

Syntax	`:CALCulate:PSUPply:MAVerage?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	The query form of this command delivers back the frequency (in Hz) and the maximum audio spectrum level of the measured audio spectrum curve. The string returned will contain two floating point real numbers. The physical dimension of the measurement result values returned is for the first value Hz and for the second value dB. **Note:** Please note that you need to have a audio measurement taken before this command will return any meaningful result. We recommend to use the `:MEAS:BLOC:AFSP:AVG` command to do so.

Example	`:MEAS:BLOC:AFSP:AVG 10`
	`:CALC:AFSP:MAXP?`
	This command will read at the audio spectrum measurement maximum the frequency and the measurement result value. The values returned in this example is: `"1230.0,-5.2"`.

:CALCulate:PSUPply:MAVerage?

Syntax	`:CALCulate:PSUPply:MAVerage?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Calculates and returns the **average** measurement result value of the current measurement started last. The string delivered back will contain as many average values as measurement types performed. The single average values will always have the format of floating point real numbers and will be separated by commas.
Example	**`:CALC:RES`** **`:MEAS:ARR:PSUP:ALL`** **`:CALC:PSUP:MAV?`** **`603.1,215.8,908.2`** Returns the average values for average power consumption (603.1 mW), average current consumption (215.8 mA) and peak current consumption (908.2 mA).

:CALCulate:PSUPply:MMINimum?

Syntax	`:CALCulate:PSUPply:MMINimum?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **minimum** measurement result value of the current measurement started last. The string delivered back will contain as many minimum values as measurement types performed. The single minimum values will always have the format of floating point real numbers and will be separated by commas.

Example	**:CALC:RES** **:MEAS:ARR:PSUP:ALL** **:CALC:PSUP:MMIN?** Returns the minimum values for average power consumption (in mW), average current consumption (in mA), and peak current consumption (in mA).

:CALCulate:PSUPply:MMAXimum?

Syntax	:CALCulate:PSUPply:MMAXimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **maximum** measurement result value of the current measurement started last. The string delivered back will contain as many maximum values as measurement types performed. The single maximum values will always have the format of floating point real numbers and will be separated by commas.
Example	**:CALC:RES** **:MEAS:ARR:PSUP:ALL** **:CALC:PSUP:MMAX?** Returns the maximum values for average power consumption (in mW), average current consumption (in mA), and peak current consumption (in mA).

:CALCulate:PSUPply:MSIGma?

Syntax	:CALCulate:PSUPply:MSIGma?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **mean value** and the **standard deviation** of the current measurement started last. The string delivered back will contain as many pairs of mean and standard deviation values as measurement types performed. All values have the format of floating point real numbers and are separated by commas.
Example	**:CALC:RES** **:MEAS:ARR:PSUP:ALL** **:CALC:PSUP:MSIG?** Returns three pairs of mean value and standard deviation for average power consumption (in mW), average current consumption (in mA), and peak current consumption (in mA).

:CALCulate:PSUPply:ALL:LIMit[:FAIL]?

Syntax	:CALCulate:PSUPply:ALL:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	This command delivers three boolean values, indicating whether the user-definable limits of the current measurements were violated by a measurement result value or not. For each of the three boolean values, a **0** means that all measurement result values were within the limits set, while a **1** indicates that at least one measurement result value violated at least one of the limits. The limits can be set using the commands described below. The results are in the sequence power consumption, average current consumption, peak current consumption.
Example	**:MEAS:ARR:PSUP:CPEA 5** **:CALC:PSUP:ALL:LIM?** String returned: **"1,0,0"** This string delivered back indicates that there was a violation of the limits for the power consumption while the other two types of measurements passed the limits.

:CALCulate:PSUPply:ALL:LIMit:STATe

Syntax	:CALCulate:PSUPply:ALL:LIMit:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches the check of the measurement result values of the MEAS:PSUP:ALL measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	**:CALC:PSUP:ALL:LIM:STAT ON** Switches the limit check for the current measurements on.

:CALCulate:PSUPply:ALL:LIMit:UPPer[:DATA]

Syntax	:CALCulate:PSUPply:ALL:LIMit:UPPer[:DATA] <real1>,<real2>,<real3>
Parameters	**real1**, **real2** and **real3** are floating point real numbers. – **real1** is the upper limit for the average power consumption. Its minimum value is **0.0**, the maximum is **2000.0**, the default value is **2000.0**. – **real2** is the upper limit for average current consumption. Its minimum value is **0.0**, the maximum is **1000.0**, the default value is **1000.0** – **real3** is the upper limit for peak current consumption. Its minimum value is **0.0**, the maximum is **4000.0**, the default value is **4000.0**
Description	Sets the **upper** limits for the current measurements.
Query	The query form of this command is not available.
Example	**:CALC:PSUP:ALL:LIM:UPP 2.3, 200.0, 1400.0** Sets the upper limit of the current measurements to 2.3 W for the average power, 200 mA for the average current and 1400 mW for the peak current.

:CALCulate:PSUPply:ALL:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:PSUPply:ALL:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>`
Parameters	**real1**, **real2**, **real3** are floating point real numbers. – **real1** is the lower limit for power consumption. Its minimum value is **0.0**, the maximum is **2000.0**, the default value is **2000.0**. – **real2** is the lower limit for average current consumption. Its minimum value is **0.0**, the maximum is **1000.0**, the default value is **1000.0** – **real3** is the lower limit for peak current consumption. Its minimum value is **0.0**, the maximum is **4000.0**, the default value is **4000.0**
Description	Sets the **lower** limits for the current measurements.
Query	The query form of this command is not available.
Example	**:CALC:PSUP:ALL:LIM:LOW 0,0,0** Sets the lower limit of the current measurements to all 0.0.

:CALCulate:PSUPply:APOWer:LIMit[:FAIL]?

Syntax	`:CALCulate:PSUPply:APOWer:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the average power consumption measurement were violated by a measurement result value or not. A 0 means that all measurement result values were within the limits set, while a 1 indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:CALC:PSUP:APOW:LIM:UPP 2.3** **:CALC:PSUP:APOWer:LIM:LOW 0** **:MEAS:ARRay:PSUP:APOWer 5** **:CALC:PSUP:APOWer:LIM?** String returned in this example:"1" This string delivered back indicates that there was a violation of the limits set.

:CALCulate:PSUPply:APOWer:LIMit:STATe

Syntax	`:CALCulate:PSUPply:APOWer:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON|OFF`. Default is `ON`.
Description	This command switches the check of the power consumption measurement result values of the `MEAS:PSUP:APOWer` measurement against their user-definable limits either `ON` or `OFF`. **Note:** All current measurements require the MS Power Supply and Current Measurement options to be installed on your Willtek 3100.

Query	There is no query form of this command available.
Example	**:CALC:PSUP:APOW:LIM:STAT ON** Switches the limit check of the current measurement on.

:CALCulate:PSUPply:APOWer:LIMit:UPPer[:DATA]

Syntax	:CALCulate:PSUPply:APOWer:LIMit:UPPer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value is 0.0, the maximum is 2000.0. The resolution is 1, the default value is 2000.0.
Description	Sets the **upper** limit for the power consumption measurement. The physical dimension of the number stated is mW.
Query	The query form of this command is not available.
Example	**:CALC:PSUP:APOW:LIM:UPP 100** Sets the upper limit of the power consumption measurement to 100 mW.

:CALCulate:PSUPply:APOWer:LIMit:LOwer[:DATA]

Syntax	:CALCulate:PSUPply:APOWer:LIMit:LOwer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value is 0.0, the maximum is 2000.0. The resolution is 1, the default value is 2000.0.
Description	Sets the **lower** limit for the power consumption measurement. The physical dimension of the number stated is mW.
Query	The query form of this command is not available.
Example	**:CALC:PSUP:APOW:LIM:LOW 10** Sets the lower limit of the power consumption measurement to 10 mW.

:CALCulate:PSUPply:ACURrent:LIMit:STATe

Syntax	:CALCulate:PSUPply:ACURrent:LIMit:STATe <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: ON|OFF. Default is ON.
Description	This command switches the check of the current measurement result values of the MEAS:PSUP:ACURrent measurement against their user-definable limits either ON or OFF. Note All current measurements require the MS Power Supply and Current Measurement options to be installed on your Willtek 3100.
Query	The query form of this command is not available.
Example	**:CALC:PSUP:ACUR:LIM:STAT ON** Switches the limit check of the average current measurement on.

:CALCulate:PSUPply:ACURrent:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:PSUPply:ACURrent:LIMit:UPPer[:DATA] <real1>`
Parameters	`real1` is a floating point real number. The minimum value is `0.0`, the maximum is `1000.0`. The resolution is `1`, the default value is `1000.0`.
Description	Sets the **upper** limit for the average current measurement. The physical dimension of the number stated is mA.
Query	The query form of this command is not available.
Example	**:CALC:PSUP:ACUR:LIM:UPP 400** Sets the upper limit of the average current measurement to 400 mA.

:CALCulate:PSUPply:ACURrent:LIMit:LOwer[:DATA]

Syntax	`:CALCulate:PSUPply:ACURrent:LIMit:LOwer[:DATA] <real1>`
Parameters	`real1` is a floating point real number. The minimum value is `0.0`, the maximum is `1000.0`. The resolution is `1`, the default value is `0.0`.
Description	Sets the **lower** limit for the peak current consumption measurement. The physical dimension of the number stated is mA.
Query	The query form of this command is not available.
Example	**:CALC:PSUP:ACUR:LIM:LOW 10** Sets the lower limit of the current consumption measurement to 10 mA.

:CALCulate:PSUPply:PCURrent:LIMit[:FAIL]?

Syntax	`:CALCulate:PSUPply:PCURrent:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers one boolean number, indicating whether the user-definable limits of the peak current measurement were violated by a measurement result value or not. A 0 means that all measurement result values were within the limits set, while a 1 indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below.
Example	**:CALC:PSUP:PCUR:LIM:UPP 1000.0** **:CALC:PSUP:PCURrent:LIM:LOW 0** **:MEAS:ARRay:PSUP:PCURrent 5** **:CALC:PSUP:PCURrent:LIM?** String returned in this example: "**1**" This string delivered back indicates that there was a violation of the defined limits.

:CALCulate:PSUPply:PCURrent:LIMit:STATe

Syntax	:CALCulate:PSUPply:PCURrent:LIMit:STATe <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: ON|OFF. Default is ON.
Description	This command switches the check of the peak current measurement result values of the MEAS:PSUP:PCURrent measurement against their user-definable limits either ON or OFF. Note All current consumption measurements require the MS Power Supply and Current Measurement options to be installed on your Willtek 3100.
Query	There is no query form of this command available.
Example	**:CALC:PSUP:PCUR:LIM:STAT ON** Switches the limit check of the peak current measurement on.

:CALCulate:PSUPply:PCURrent:LIMit:UPPer[:DATA]

Syntax	:CALCulate:PSUPply:PCURrent:LIMit:UPPer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value is 0.0, the maximum is 4000.0. The resolution is 1, the default value is 4000.0.
Description	Sets the **upper** limit for the peak current consumption measurement. The physical dimension of the number stated is mA.
Query	There is no query form of this command available.
Example	**:CALC:PSUP:PCUR:LIM:UPP 1000.0** Sets the upper limit of the peak current consumption measurement to 1000 mA.

:CALCulate:PSUPply:PCURrent:LIMit:LOwer[:DATA]

Syntax	:CALCulate:PSUPply:PCURrent:LIMit:LOwer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value is 0.0, the maximum is 4000.0. The resolution is 1, the default value is 0.0.
Description	Sets the **lower** limit for the peak current consumption measurement. The physical dimension of the number stated is mA.
Query	There is no query form of this command available.
Example	:CALC:PSUP:PCUR:LIM:LOW 10 Sets the lower limit of the peak current consumption measurement to 10 mA.

:CALCulate:GSM :RFTX:CORNer:RACH:LIMit[:FAIL]?

Syntax	:CALC:GPRS :RFTX:CORNer:RACH:LIMit[:FAIL]?
Parameters	There are no parameters.

Description	There is solely a query form of this command available.
Query	This command delivers eight boolean numbers, indicating whether the limits for the eight corner positions of **random access** burst were violated by the bursts received or not. A `0` on a position means that there was no violation of the limits while a `1` indicates that at least one measurement result did violate one of the two corresponding measurement limits. The eight boolean numbers correspond with the eight corner points set with the `:CALC:GPRS:RFTX:CORN:RACH:POS` command as described above. Please refer to section Corner Points for further details.
Example	`:CALC:GPRS:RFTX:CORN:RACH:LIM:FAIL?` String returned:"`0,1,0,0,0,0,0,0`" This string indicates that one of the two limits set for corner point 2 was violated.

:CALCulate:GSM:RFTX:CORN:RACH:LIMit:STATe

Syntax	`:CALCulate:GSM:RFTX:CORN:RACH:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON|OFF`. Default is `ON`.
Description	This command switches the check of the bursts against the limits at the corner points either on or off for **random access** bursts.
Query	There is no query form of this command available.
Example	`:CALC:GPRS:RFTX:CORN:RACH:LIM:STAT OFF` Switches the limit check at the eight corner positions off.

:CALCulate:GSM:RFTX:CORNer:RACH:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GSM`
	`:RFTX:CORNer:RACH:LIMit:UPPer[:DATA]`
	`<real1>,<real2>,<real3>,<real4>,`
	`<real5>,<real6>,<real7>,<real8>`

Parameters	`realx` are floating point real numbers. The minimum value for all `realx` is `-150.0`, the maximum is `10.0`. The minimum resolution possible for all `realx` is `0.01`. The default value is `4.0` for all `realx`.
Description	Sets the **upper** limits at the eight user-definable corner points for **random access** bursts. The single data values are dB relative to the power level during the active part of the burst. For further details, please refer to section Corner Points.
Query	There is no query form to this command.
Example	`:CALC:GSM:RFTX:CORN:RACH:LIM:UPP 4,4,4,4,4,4,4,4`

:CALCulate:GSM :RFTX:CORNer:RACH:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GSM :RFTX:CORNer:RACH:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>
Parameters	realx are floating point real numbers. The minimum value for all realx is –150.0, the maximum is 10.0. The minimum resolution possible for all realx is 0.01. The default value is –150.0 for all realx.
Description	Sets the **lower** limits at the eight user-definable corner points for **random access** bursts. The single data values are dB relative to the power level during the active part of the burst. For further details, please refer to section Corner Points.
Query	There is no query form to this command.
Example	:CALC:GSM:RFTX:CORN:RACH:LIM:LOW –150,–150,–150,–150,–150,–150,–150,–150

:CALCulate:WCDMa:RFTX:MAVerage?

Syntax	:CALCulate:WCDMa:RFTX:MAVerage?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	There is a query form which returns the stored settings.
Example	:MEAS:WCDM:RFTX:FREQ :CALC:RES ...(other SCPI commands)... :CALC:WCDM:RFTX:MAV? In this example, first a continuous measurement of the mobile's frequency error is started. After some time, the :CALC:WCDM:RFTX:MAV command is used to read out the average frequency deviation of the mobile. The string returned in this example is "12.9".

:CALCulate:WCDMa:RFTX:MMINimum?

Syntax	:CALCulate:WCDMa:RFTX:MMINimum?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **minimum** measurement result value of the TX measurement started last. The string delivered will contain as many minimum values as measurement types performed. The single minimum values will always have the format of floating point real numbers and will be separated by commas.

Example	`:MEAS:WCDM:RFTX:FREQ`
	`:CALC:RES`
	`...(other SCPI commands)...`
	`:CALC:WCDM:RFTX:MMIN?`
	In this example, first a continuous measurement of the mobile's frequency error is started. After some time, the `:CALC:WCDM:RFTX:MMAX` command is used to read out the minimum frequency deviation of the mobile. The string returned in this example is `"10.2"`.

:CALCulate:WCDMa:RFTX:MMAXimum?

Syntax	`:CALCulate:WCDMa:RFTX:MMAXimum?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **maximum** measurement result value of the TX measurement started last. The string delivered will contain as many maximum values as measurement types performed. The single maximum values will always have the format of floating point real numbers and will be separated by commas.

Example	`:MEAS:WCDM:RFTX:FREQ`
	`:CALC:RES`
	`...(other SCPI commands)...`
	`:CALC:WCDM:RFTX:MMAX?`
	In this example, first a continuous measurement of the mobile's frequency error is started. After some time, the `:CALC:WCDM:RFTX:MMAX` command is used to read out the maximum frequency deviation of the mobile. The string returned in this example is `"22.1"`.

:CALCulate:WCDMa:RFTX:MSIGma?

Syntax	`:CALCulate:WCDMa:RFTX:MSIGma?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	Returns the **mean value** and the **standard deviation** of the TX measurement last started. The string delivered contains as many pairs of mean and standard deviation values as measurement types performed. The individual values all have the format of floating point real numbers and will be separated by commas.

Example	`:MEAS:WCDM:RFTX:FREQ`
	`:CALC:RES`
	`...(other SCPI commands)...`
	`:CALCulate:WCDM:RFTX:MSIG?`
	In this example, first a continuous measurement of the mobile's frequency error is started. After some time, the `:CALC:WCDM:RFTX:MSIG` command is used to read out the mean frequency error and its standard deviation. The string returned in this example is `"15.7,3.4"`.

:CALCulate:WCDMa:RFTX:FREQ:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:FREQ:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFTX:FREQ` measurement violated its limits.

Example	`:MEAS:WCDMA:ARR:RFTX:FREQ 20`
	`:CALC:WCDM:RFTX:FREQ:LIM?`
	The `MEASurement` command starts 20 frequency measurements. The query returns `1` if any of the measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFTX:FREQuency:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:FREQuency:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON | OFF`. The default is `ON`.
Description	This command switches the check of the measurement result values of the measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFTX:FREQ:LIM:STAT ON` Switches the limit check for the frequency error measurement on.

:CALCulate:WCDMa:RFTX:FREQuency:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:FREQuency:LIMit:UPPer:[DATA] <real1>`
Parameters	`int1` is a real number. The minimum value for `real1` is `0`, the maximum is `1000`, the default is `200`.
Description	This command sets the upper limit of the frequency error measurement by band. The physical dimension of the number stated is Hz.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:FREQ:LIM:UPP 310`
	`:CALC:WCDM:RFTX:FREQ:LIM:UPP?`
	The query returns the previously set limit of `310` (hertz).

:CALCulate:WCDMa:RFTX:FREQuency:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:FREQuency:LIMit:LOWer:[DATA] <real1>`
Parameters	`int1` is a real number. The minimum value for `real1` is `-1000`, the maximum is `0`, the default is `-200`.

Description	This command sets the lower limit of the frequency error measurement by band. The physical dimension of the number stated is Hz.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:FREQ:LIM:LOW -310`
	`:CALC:WCDM:RFTX:FREQ:LIM:LOW?`
	The query returns the previously set limit of `-310` (hertz).

:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFTX:POW:MEAN` measurement violated the defined limits.

Example	`:MEAS:WCDMA:ARR:RFTX:POW:MEAN 20`
	`:CALC:WCDM:RFTX:POW:MEAN:LIM?`
	The query returns `1` if any of the 20 measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` can take on one of the following values: `ON|OFF`. The default is `ON`.
Description	This command switches the check of the measurement result values of the mean power measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFTX:POW:MEAN:LIM:STAT ON` Switches the limit check for the frequency error measurement on.

:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit:UPPer:[DATA] <real>`
Parameters	`real` is a floating point real number. Its minimum value is `-120.0`, the maximum is `50.0`, the resolution is `0.1` and the default is `34.0`.
Description	Sets the upper limit for the mobile's RF output mean power. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:POW:MEAN:LIM:UPP 12.5`
	`:CALC:WCDM:RFTX:POW:MEAN:LIM:UPP?`
	Returns `12.5`.

:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit:LOWer:[DATA]

Syntax	:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit:LOWer:[DATA] <real>
Parameters	real is a floating point real number. Its minimum value is –120.0, the maximum is 50.0, the resolution is 0.1 and the default is –60.0.
Description	Sets the lower limit for the mobile's RF output mean power. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFTX:POW:MEAN:LIM:LOW –32.5 :CALC:WCDM:RFTX:POW:MEAN:LIM:LOW? Returns –32.5.

:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit:MAXimum :UPPer:[DATA]

Syntax	:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit :MAXimum:UPPer:[DATA] <Real1>,<Real2>,<Real3>,<Real4>
Parameters	realx are floating point real numbers. The minimum value of <Real1> is 10.0, the maximum is 44.0 and the default is 34.0. The minimum value of <Real2> is 4.0, the maximum is 38.0 and the default is 28.0. The minimum value of <Real3> is 1.0, the maximum is 35.0 and the default is 25.0. The minimum value of <Real4> is –1.0, the maximum is 33.0 and the default is 23.0. The resolution of all values is 0.1.
Description	Sets the upper limits for the mobile's RF output mean maximum power for the power classes 1, 2, 3 and 4. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFTX:POW:MEAN:LIM:MAX:UPP 35,29,26,24 :CALC:WCDM:RFTX:POW:MEAN:MAX:LIM:UPP? Returns 35.0,29.0,26.0,24.0.

:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit:MAXimum :LOWer:[DATA]

Syntax	:CALCulate:WCDMa:RFTX:POWer:MEAN:LIMit:MAXi-mum:LOWer:[DATA] <Real1>,<Real2>,<Real3>,<Real4>
Parameters	realx are floating point real numbers. The minimum value of <Real1> is 10.0, the maximum is 44.0 and the default is 30.0. The minimum value of <Real2> is 4.0, the maximum is 38.0 and the default is 24.0. The minimum value of <Real3> is 1.0, the maximum is 35.0 and the default is 21.0. The minimum value of <Real4> is –1.0, the maximum is 33.0 and the default is 19.0. The resolution of all values is 0.1.
Description	Sets the lower limits for the mobile's RF output mean maximum power for the power classes 1, 2, 3 and 4. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.

Example	:CALC:WCDM:RFTX:POW:MEAN:LIM:MAX:LOW 29,23,20,18 :CALC:WCDM:RFTX:POW:MEAN:MAX:LIM:LOW? Returns 29.0,23.0,20.0,18.0.

:CALCulate:WCDMa:RFTX:POWer:MINimum:LIMit:UPPer:[DATA]

Syntax	:CALCulate:WCDMa:RFTX:POWer:MINimum:LIMit:UPPer:[DATA] <real>
Parameters	real is a floating point real number. Its minimum value is –120.0, the maximum is 0.0, the resolution is 0.1 and the default is –50.0.
Description	Sets the upper limit for the mobile's RF mean minimum power. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFTX:POW:MIN:LIM:UPP –40.5 :CALC:WCDM:RFTX:POW:MIN:LIM:UPP? Returns –40.5.

:CALCulate:WCDMa:RFTX:POWer:MINimum:LIMit:LOWer:[DATA]

Syntax	:CALCulate:WCDMa:RFTX:POWer:MINimum:LIMit:LOWer:[DATA] <real>
Parameters	real is a floating point real number. Its minimum value is –120.0, the maximum is 0.0, the resolution is 0.1 and the default is –120.0.
Description	Sets the lower limit for the mobile's RF mean minimum power. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFTX:POW:MIN:LIM:LOW –60.5 :CALC:WCDM:RFTX:POW:MIN:LIM:LOW? Returns –60.5.

CALCulate:WCDMa:RFTX:POWer:PEAK:LIMit[:FAIL]

Syntax	:CALCulate:WCDMa:RFTX:POWer:PEAK:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the :MEAS:WCDMA:ARR:RFTX:POW:PEAK measurement violated the defined limits.
Example	:MEAS:WCDMA:ARR:RFTX:PEAK:POW 20 :CALC:WCDM:RFTX:POW:PEAK:LIM? Returns 0 if any of the 20 power measurements was out of limits, and 1 if all results were within limits.

:CALCulate:WCDMa:RFTX:POWer:PEAK:LIMit:STATe

Syntax	:CALCulate:WCDMa:RFTX:POWer:PEAK:LIMit:STATe <PredefExp>
Parameters	PredefExp can take on one of the following values: ON|OFF. The default is ON.
Description	This command switches the check of the measurement result values of the peak power measurement against their user-definable limits either ON or OFF.
Query	There is a query form whicht returns the sotred settings.
Example	:CALC:WCDM:RFTX:POWer:PEAK:LIM:STAT ON Switches the limit check for the frequency error measurement on

:CALCulate:WCDMa:RFTX:POWer:PEAK:LIMit:UPPer:[DATA]

Syntax	:CALCulate:WCDMa:RFTX:POWer:PEAK:LIMit:UPPer:[DATA] <real>
Parameters	real is a floating point real number. Its minimum value is -120.0, the maximum is 50.0, the resolution is 0.1 and the default is 34.0.
Description	Sets the upper limit for the mobile's RF output peak power. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFTX:POW:PEAK:LIM:UPP 2.5 :CALC:WCDM:RFTX:POW:PEAK:LIM:UPP? Returns 2.5.

:CALCulate:WCDMa:RFTX:POWer:PEAK:LIMit:LOWer:[DATA]

Syntax	:CALCulate:WCDMa:RFTX:POWer:PEAK:LIMit:LOWer:[DATA] <real>
Parameters	real is a floating point real number. Its minimum value is -120.0, the maximum is 50.0, the resolution is 0.1 and the default is -60.0.
Description	Sets the lower limit for the mobile's RF output peak power. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFTX:POW:PEAK:LIM:LOW -2.5 :CALC:WCDM:RFTX:POW:PEAK:LIM:LOW? Returns -2.5.

:CALCulate:WCDMa:RFTX:CPOWer:LIMit[:FAIL]

Syntax	:CALCulate:WCDMa:RFTX:CPOWer:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	This command delivers a boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFTX:CPOW` measurement violated the defined limits.

Example	`:MEAS:WCDMA:ARR:RFTX:CPOW 20`
	`:CALC:WCDM:RFTX:CPOW:LIM?`
	Returns `1` if any of the 20 power measurements was out of limits, and `0` if all results were inside limits.

:CALCulate:WCDMa:RFTX:CPOWer:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:CPOWer:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` can take on one of the following values: `ON|OFF`. The default is `ON`.
Description	This command switches the check of the measurement result values of the channel power measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFTX:CPOW:LIM:STAT ON` Switches the limit check for the frequency error measurement on.

:CALCulate:WCDMa:RFTX:CPOWer:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:CPOWer:LIMit:UPPer:[DATA] <real>`
Parameters	`real` is a floating point real number. Its minimum value is `-120.0`, the maximum is `50.0`, the resolution is `0.1` and the default is `34.0`.
Description	Sets the upper limit for the mobile's RF output channel power. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:CPOW:LIM:UPP 2.5`
	`:CALC:WCDM:RFTX:CPOW:LIM:UPP?`
	Returns `2.5`.

:CALCulate:WCDMa:RFTX:CPOWer:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:CPOWer:LIMit:LOWer:[DATA] <real>`
Parameters	`real` is a floating point real number. Its minimum value is `-120.0`, the maximum is `50.0`, the resolution is `0.1` and the default is `-60.0`.
Description	Sets the lower limit for the mobile's RF output channel power. The physical dimension of the number stated is dBm.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:CPOW:LIM:LOW -2.5`
	`:CALC:WCDM:RFTX:CPOW:LIM:LOW?`
	Returns `-2.5`.

:CALCulate:WCDMa:RFTX:MODQuality:ALL:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:ALL:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFTX:MODQ:ALL` measurement violated its limits.

Example	`:MEAS:WCDMA:ARR:RFTX:MODQ:ALL 20`
	`:CALC:WCDM:RFTX:MODQ:ALL:LIM?`
	The query returns `1` if any of the measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFTX:MODQuality:ALL:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:ALL:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` may take on one of the following predefined expressions: `ON|OFF`. The default is `ON`.
Description	This command switches the check of the measurement result values of all modulation quality measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFTX:MODQ:ALL:LIM:STAT ON` Switches the limit check for all modulation quality measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:ERMS:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:ERMS:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFTX:MODQ:ERMS` measurement violated its limits.

Example	`:MEAS:WCDMA:ARR:RFTX:MODQ:ERMS 20`
	`:CALC:WCDM:RFTX:MODQ:ERMS:LIM?`
	Returns `1` if any of the EVM RMS vector error measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFTX:MODQuality:ERMS:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:ERMS:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` may take on one of the following predefined expressions: `ON|OFF`. The default is `ON`.

Description	This command switches the check of the measurement result values of the EVM RMS vector error measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFTX:MODQ:ERMS:LIM:STAT ON` Switches the limit check for the EVM RMS-averaged vector error measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:ERMS:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:ERMS:LIMit:UPPer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for the parameter is `0.0`, the maximum is `200.0`, the resolution is `0.1`, the default is value is `17.5`.
Description	Sets the upper limit for the EVM RMS average vector error. The physical dimension of the number stated is percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:ERMS:LIM:UPP 200`
	`:CALC:WCDM:RFTX:MODQ:ERMS:LIM:UPP?`
	The query returns the previously set value `200.0`.

:CALCulate:WCDMa:RFTX:MODQuality:ERMS:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:ERMS:LIMit:LOWer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for the parameter is `0.0`, the maximum is `200.0`, the resolution is `0.1`, the default is value is `0.0`.
Description	Sets the lower limit for the EVM RMS average vector error. The physical dimension of the number stated is percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:ERMS:LIM:LOW 200`
	`:CALC:WCDM:RFTX:MODQ:ERMS:LIM:LOW?`
	The query returns the previously set value `200.0`.

:CALCulate:WCDMa:RFTX:MODQuality:EPEAk:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:EPEAk:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFTX:MODQ:EPEA` measurement violated its limits.

Example	`:MEAS:WCDMA:ARR:RFTX:MODQ:EPEA 20`
	`:CALC:WCDM:RFTX:MODQ:EPEA:LIM?`
	Returns `0` if all 20 results are within limits, or `1` if any result is outside the limits.

:CALCulate:WCDMa:RFTX:MODQuality:EPEAk:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:EPEAk:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON|OFF`. Default is `ON`.
Description	This command switches the check of the measurement result values of the measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFTX:MODQ:EPEA:LIM:STAT ON` Switches the limit check for the EVM peak vector error measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:EPEAk:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:EPEAk:LIMit:UPPer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. Its minimum value is `0.0`, the maximum is `200.0`, the resolution is `0.1` and the default is value is `50.0`.
Description	Sets the upper limit for the EVM peak vector error measurement. The physical dimension of the number is stated as a percentage.
Query	There is query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:EPEA:LIM:UPP 200`
	`:CALC:WCDM:RFTX:MODQ:EPEA:LIM:UPP?`
	Returns `200`.

:CALCulate:WCDMa:RFTX:MODQuality:EPEAk:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:EPEAk:LIMit:LOWer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. Its minimum value is `0.0`, the maximum is `200.0`, the resolution is `0.1`, the default is value is `0.0`.
Description	Sets the lower limit for the EVM peak vector error measurement. The physical dimension of the number is stated as a percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:EPEA:LIM:LOW 0`
	`:CALC:WCDM:RFTX:MODQ:EPEA:LIM:LOW?`
	Returns `0` in this example.

:CALCulate:WCDMa:RFTX:MODQuality:MRMS:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:MRMS:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFTX:MODQ:MRMS` measurement violated its limits.

Example	`:MEAS:WCDMA:ARR:RFTX:MODQ:MRMS 20`
	`:CALC:WCDM:RFTX:MODQ:MRMS:LIM?`
	The query returns `1` if any of the measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFTX:MODQuality:MRMS:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:MRMS:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` may take on one of the following predefined expressions: `ON|OFF`. The default is `ON`.
Description	This command switches the check of the measurement result values of the magnitude RMS vector error measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFTX:MODQ:MRMS:LIM:STAT ON` Switches the limit check for the magnitude RMS-averaged vector error measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:MRMS:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:MRMS:LIMit:UPPer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for parameter is `0.0`, the maximum is `200.0`, the resolution is `0.1`, the default is value is `17.5`.
Description	Sets the upper limit for the magnitude RMS average vector error. The physical dimension of the number stated is percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:MRMS:LIM:UPP 200`
	`:CALC:WCDM:RFTX:MODQ:MRMS:LIM:UPP?`
	The query returns the previously set value `200.0`.

:CALCulate:WCDMa:RFTX:MODQuality:MRMS:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:MRMS:LIMit:LOWer:[DATA] <Real1>`

Parameters	`Real1` is a floating point real number. The minimum value for parameter is `0.0`, the maximum is `200.0`, the resolution is `0.1`, the default is value is `0.0`.
Description	Sets the lower limit for the magnitude RMS average vector error. The physical dimension of the number stated is percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:MRMS:LIM:LOW 200`
	`:CALC:WCDM:RFTX:MODQ:MRMS:LIM:LOW?`
	The query returns the previously set value `200.0`.

:CALCulate:WCDMa:RFTX:MODQuality:MPEAk:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:MPEAk:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFTX:MODQ:MPEA` measurement violated its limits.

Example	`:MEAS:WCDMA:ARR:RFTX:MODQ:MPEA 20`
	`:CALC:WCDM:RFTX:MODQ:MPEA:LIM?`
	Returns `0` if all 20 results are within limits, or `1` if any result is outside the limits.

:CALCulate:WCDMa:RFTX:MODQuality:MPEAk:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:MPEAk:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON|OFF`. Default is `ON`.
Description	This command switches the check of the measurement result values of the measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFTX:MODQ:MPEA:LIM:STAT ON` Switches the limit check for the magnitude peak vector error measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:MPEAk:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:MPEAk:LIMit:UPPer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. Its minimum value is `0.0`, the maximum is `200.0`, the resolution is `0.1`, and the default is value is `50.0`.
Description	Sets the upper limit for the magnitude peak vector error measurement. The physical dimension of the number is stated as a percentage.
Query	There is a query form which returns the stored settings.

Example	:CALC:WCDM:RFTX:MODQ:MPEA:LIM:UPP 200 :CALC:WCDM:RFTX:MODQ:MPEA:LIM:UPP? Returns 200.

:CALCulate:WCDMa:RFTX:MODQuality:MPEAk:LIMit:LOWer:[DATA]

Syntax	:CALCulate:WCDMa:RFTX:MODQuality:MPEAk:LIMit:LOWer:[DATA] <Real1>
Parameters	Real1 is a floating point real number. Its minimum value is 0.0, the maximum is 200.0, the resolution is 0.1, the default is value is 0.0.
Description	Sets the lower limit for the magnitude peak vector error measurement. The physical dimension of the number is stated as a percentage.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFTX:MODQ:MPEA:LIM:LOW 0 :CALC:WCDM:RFTX:MODQ:MPEA:LIM:LOW? Returns 0 in this example.

:CALCulate:WCDMa:RFTX:MODQuality:PRMS:LIMit[:FAIL]

Syntax	:CALCulate:WCDMa:RFTX:MODQuality:PRMS:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the :MEAS:WCDMA:ARR:RFTX:MODQ:PRMS measurement violated its limits.
Example	:MEAS:WCDMA:ARR:RFTX:MODQ:PRMS 20 :CALC:WCDM:RFTX:MODQ:PRMS:LIM? The query returns 1 if any of the measurement results was out of limits, or 0 if none of the results was out of limits.

:CALCulate:WCDMa:RFTX:MODQuality:PRMS:LIMit:STATe

Syntax	:CALCulate:WCDMa:RFTX:MODQuality:PRMS:LIMit:STATe <PredefExp>
Parameters	PredefExp may take on one of the following predefined expressions: ON|OFF. The default is ON.
Description	This command switches the check of the measurement result values of the Phase RMS vector error measurement against their user-definable limits either ON or OFF.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFTX:MODQ:PRMS:LIM:STAT ON Switches the limit check for the Phase RMS-averaged vector error measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:PRMS:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:PRMS:LIMit:UPPer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for parameter is `0.0`, the maximum is `200.0`, the resolution is `0.1`, the default is value is `10.0`.
Description	Sets the upper limit for the Phase RMS average vector error. The physical dimension of the number stated is percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:PRMS:LIM:UPP 200`
	`:CALC:WCDM:RFTX:MODQ:PRMS:LIM:UPP?`
	The query returns the previously set value `200.0`.

:CALCulate:WCDMa:RFTX:MODQuality:PRMS:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:PRMS:LIMit:LOWer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for parameter is `0.0`, the maximum is `200.0`, the resolution is `0.1`, the default is value is `0.0`.
Description	Sets the lower limit for the Phase RMS average vector error. The physical dimension of the number stated is percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:PRMS:LIM:LOW 200`
	`:CALC:WCDM:RFTX:MODQ:PRMS:LIM:LOW?`
	The query returns the previously set value `200.0`.

:CALCulate:WCDMa:RFTX:MODQuality:PPEAk:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:PPEAk:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFTX:MODQ:PPEA` measurement violated its limits.

Example	`:MEAS:WCDMA:ARR:RFTX:MODQ:PPEA 20`
	`:CALC:WCDM:RFTX:MODQ:PPEA:LIM?`
	Returns `0` if all 20 results are within limits, or `1` if any result is outside the limits.

:CALCulate:WCDMa:RFTX:MODQuality:PPEAk:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:PPEAk:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON|OFF`. Default is `ON`.

Description	This command switches the check of the measurement result values of the measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFTX:MODQ:PPEA:LIM:STAT ON` Switches the limit check for the phase peak vector error measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:PPEAk:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:PPEAk:LIMit:UPPer:[DATA]`
Parameters	`Real1` is a floating point real number. Its minimum value is `0.0`, the maximum is `200.0`, the resolution is `0.1`, and the default is value is `45.0`.
Description	Sets the upper limit for the phase peak vector error measurement. The physical dimension of the number is stated as a percentage.
Query	There is query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:PPEA:LIM:UPP 200`
	`:CALC:WCDM:RFTX:MODQ:PPEA:LIM:UPP?`
	Returns `200`.

:CALCulate:WCDMa:RFTX:MODQuality:PPEAk:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:PPEAk:LIMit:LOWer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. Its minimum value is `0.0`, the maximum is `200.0`, the resolution is `0.1`, the default is value is `0.0`.
Description	Sets the lower limit for the Phase peak vector error measurement. The physical dimension of the number is stated as a percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:PPEA:LIM:LOW 0`
	`:CALC:WCDM:RFTX:MODQ:PPEA:LIM:LOW?`
	Returns `0` in this example.

:CALCulate:WCDMa:RFTX:MODQuality:RHO:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:RHO:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the waveform quality measurement exceeds the limits.

Example	`:MEAS:WCDMA:ARR:RFTX:MODQ:RHO? 1`
	`:CALC:WCDM:RFTX:MODQ:RHO:LIM?`
	The query returns `1` if any of the measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFTX:MODQuality:RHO:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:RHO:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON|OFF`. Default is `ON`.
Description	This command switches the check of the measurement result values of the waveform quality measurement against their user-definable limits either `ON` or `OFF`.
Query	There is no query form of this command available.
Example	`:CALC:WCDM:RFTX:MODQ:RHO:LIM:STAT ON` Switches the limit check for the waveform quality measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:RHO:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:RHO:LIMit:UPPer:[DATA] <PredefExp>`
Parameters	`PredefExp` is a floating point real number. The minimum value for this parameter is `0.90`, the maximum is `1.0`, the resolution is `0.0001`, the default is value is `1.0`.
Description	Sets the upper limit for the waveform quality measurement.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:RHO:LIM:UPP 1.0`
	`:CALC:WCDM:RFTX:MODQ:RHO:LIM:UPP?`
	Returns `1.0`.

:CALCulate:WCDMa:RFTX:MODQuality:RHO:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:RHO:LIMit:LOWer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for this parameter is `0.0`, the maximum is `1.0`, the resolution is `0.0001`, the default is value is `0.944`.
Description	Sets the lower limit for the waveform quality measurement.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:RHO:LIM:LOW 0.944`
	`:CALC:WCDM:RFTX:MODQ:RHO:LIM:LOW?`
	Returns `0.944`.

:CALCulate:WCDMa:RFTX:MODQuality:IQOFfset:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:IQOFfset:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the origin offset measurement exceeds the limits.

Example	`:MEAS:WCDMA:ARR:RFTX:MODQ:IQOF? 20`
	`:CALC:WCDM:RFTX:MODQ:IQOF:LIM?`
	The query returns `1` if any of the measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFTX:MODQuality:IQOFfset:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:IQOFfset:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON|OFF`. The default is `ON`.
Description	This command switches the check of the measurement result values of the origin offset measurement against their user-definable limits either `ON` or `OFF`.
Query	There is no query form of this command available.
Example	`:CALC:WCDM:RFTX:MODQ:IQOF:LIM:STAT ON` Switches the limit check for the origin offset measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:IQOFfset:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:IQOFfset:LIMit:UPPer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for parameter is `0.0`, the minimum is `0.0`, the resolution is `0.1`, the default is value is `0.0`.
Description	Sets the upper limit for the origin offset measurement. The physical dimension of the number stated is dBc.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:IQOF:LIM:UPP 0`
	`:CALC:WCDM:RFTX:MODQ:IQOF:LIM:UPP?`
	Returns `0` in this case.

:CALCulate:WCDMa:RFTX:MODQuality:IQOFfset:LIMit :LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:IQOFfset:LIMit:LOWer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. Its minimum value is `-99.0`, the maximum is `0.0`, the resolution is `0.1`, the default is value is `-99.0`.
Description	Sets the lower limit for the origin offset measurement. The physical dimension of the number stated is dBc.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:IQOF:LIM:LOW -60.0`
	`:CALC:WCDM:RFTX:MODQ:IQOF:LIM:LOW?`
	Returns `-60.0`.

:CALCulate:WCDMa:RFTX:MODQuality:IQIMbalance:LIMit [:FAIL]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:IQIMbalance:LIMit [:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the IQ imbalance measurement exceeds the limits.

Example	`:MEAS:WCDMA:ARR:RFTX:MODQ:IQIM? 20`
	`:CALC:WCDM:RFTX:MODQ:IQIM:LIM?`
	The query returns `1` if any of the measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFTX:MODQuality:IQIMbalance:LIMit :STATe

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:IQIMbalance:LIMit :STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON | OFF`. Default is `ON`.
Description	This command switches the check of the measurement result values of the IQ imbalance measurement against their user-definable limits either `ON` or `OFF`.
Query	There is no query form of this command available.
Example	`:CALC:WCDM:RFTX:MODQ:IQIM:LIM:STAT ON` Switches the limit check for the IQ imbalance measurement on.

:CALCulate:WCDMa:RFTX:MODQuality:IQIMbalance:LIMit :UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:IQIMbalance:LIMit :UPPer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for parameter is `0.0`, the maximum is `0.0`, the resolution is `0.1`, the default is value is `0.0`.
Description	This command sets the upper limit for the IQ imbalance measurement. The physical dimension of the number stated is dB.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:IQIM:LIM:UPP 0`
	`:CALC:WCDM:RFTX:MODQ:IQIM:LIM:UPP?`
	Returns `0`.

:CALCulate:WCDMa:RFTX:MODQuality:IQIMbalance:LIMit :LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFTX:MODQuality:IQIMbalance:LIMit :LOWer:[DATA] <PredefExp>`
Parameters	`PredefExp` is a floating point real number. The minimum value is `-99.0`, the maximum is `0.0`, the resolution is `0.1`, the default is value is `-99.0`.
Description	This command sets the lower limit for the IQ imbalance measurement. The physical dimension of the number stated is dB.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFTX:MODQ:IQIM:LIM:LOW -99.0`
	`:CALC:WCDM:RFTX:MODQ:IQIM:LIM:LOW?`
	Returns `-99.0`.

:CALCulate:WCDMa:RFSPectrum:MSPectrum:VALue

Syntax	`:CALCulate:WCDMa:RFSPectrum:MSPectrum:VALue <real1>`
Parameters	`real1` is a floating point real number. The minimum value is `-5,000,000.0`, the maximum is `5,000,000.0`. The minimum resolution possible is `0.1`.
Description	There is only a query form of this command available.
Query	The query form of this command delivers an (interpolated) measurement result value of the modulation spectrum curve at the frequency specified with the `real1` parameter. `real1` is interpreted as an offset from the carrier (i.e. the transmission frequency). The string returned will contain one floating point real number. The physical dimension of the measurement result value returned is dBc. **Notes** – Please note that you need to have a spectrum measurement taken before this command will return any meaningful result. We recommend to use the `:MEAS:WCDM:RFSP:MSP:AVG` command to do so. – The `:CALC:WCDM:RFSP:MSP:VAL` command is not really a measurement - it is an interpolation of the spectrum curve for the frequency specified.

Example	`:MEAS:WCDM:RFSP:MSP:AVG 20`
	`:CALC:WCDM:RFSP:MSP:VAL? -50000`
	This command will interpolate a modulation spectrum measurement at an offset of 50 kHz to the carrier. The value returned in this example is: `"-4.7"`.

:CALCulate:WCDMa:RFSPectrum:MSPectrum:MAXPeak

Syntax	`:CALCulate:WCDMa:RFSPectrum:MSPectrum:MAXPeak?`
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	The query form of this command delivers the frequency offset (in Hz) from the carrier and the maximum value of the measured modulation spectrum curve. The string returned will contain two floating point real numbers. The physical dimension of the measurement result values returned is for the first value Hz and for the second value dBc. Notes – Please note that you need to have a spectrum measurement taken before this command will return any meaningful result. We recommend to use the `:MEAS:WCDM:RFSP:MSP:AVG` command to do so. – The `:CALC:WCDM:RFSP:MSP:MAXP` command is not really a measurement - it is an readout of the spectrum curve.

Example	`:MEAS:WCDM:RFSP:MSP:AVG 20`
	`:CALC:WCDM:RFSP:MSP:MAXP?`
	This command will read at the modulation spectrum measurement maximum the frequency offset to the carrier and the measurement result value. The values returned in this example is: `"-67700.0,-5.2"`.

:CALCulate:WCDMa:RFSPectrum:MSPectrum:OBW:LIMit [:FAIL]

Syntax	`:CALCulate:WCDMa:RFSPectrum:MSPectrum:OBW:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the occupied bandwidth measurement exceeds the limits.

Example	`:MEAS:WCDMA:ARR:RFSP:MSP:OBW? 20`
	`:CALC:WCDM:RFSP:MSP:OBW:LIM?`
	The query returns `1` if any of the measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFSPectrum:MSPectrum:OBW:LIMit :STATe

Syntax	`:CALCulate:WCDMa:RFSPectrum:MSPectrum:OBW:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON|OFF`. Default is `ON`.

Description	This command switches the check of the measurement result values of the occupied bandwidth measurement against their user-definable limits either `ON` or `OFF`.
Query	There is no query form of this command available.
Example	`:CALC:WCDM:RFSP:MSP:OBW:LIM:STAT ON` Switches the limit check for the occupied bandwidth measurement on.

:CALCulate:WCDMa:RFSPectrum:MSPectrum:OBW:LIMit :UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFSPectrum:MSPectrum:OBW:LIMit:UPPer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for the parameter is `0`, the maximum is `10000000`, the resolution is `1`, the default value is `5000000`.
Description	This command sets the upper limit for the occupied bandwidth measurement. The physical dimension of the number stated is Hz.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFSP:MSP:OBW:LIM:UPP 0`
	`:CALC:WCDM:RFSP:MSP:OBW:LIM:UPP?`
	Returns `0`.

:CALCulate:WCDMa:RFSPectrum:MSPectrum:OBW:LIMit :LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFSPectrum:MSPectrum:OBW:LIMit:LOWer:[DATA] <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for parameter is `0`, the maximum is `10,000,000`, the resolution is `1`, the default value is `15000`.
Description	This command sets the lower limit for the occupied bandwidth measurement. The physical dimension of the number stated is Hz.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFSP:MSP:OBW:LIM:LOW -99.0`
	`:CALC:WCDM:RFSP:MSP:OBW:LIM:LOW?`
	Returns `-99.0`.

:CALCulate:WCDMa:RFSPectrum:ACLR:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFSPectrum:ACLR:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the adjacent channel leakage power ratio measurement exceeds the limits.

Example	`:MEAS:WCDMA:ARR:RFSP:ACLR? 20`
	`:CALC:WCDM:RFSP:ACLR:LIM?`
	The query returns `1` if any of the measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFSPectrum:ACLR:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFSPectrum:ACLR:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON | OFF`. Default is `ON`.
Description	This command switches the check of the measurement result values of the adjacent channel leakage power ratio measurement against their user-definable limits either `ON` or `OFF`.
Query	There is no query form of this command available.
Example	`:CALC:WCDM:RFSP:MSP:ACLR:STAT ON` Switches the limit check for the adjacent channel leakage power ratio measurement on.

:CALCulate:WCDMa:RFSPectrum:ACLR:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFSPectrum:ACLR:LIMit:UPPer:[DATA] <Real1>,<Real2>,<Real3>`
Parameters	`Realx` are floating point real numbers. The minimum value for the parameters is `-120.0`, the maximum is `0.0`, the resolution is `0.1`, the default value is for `Real1` `-33.0`, for `Real2` `-43.0` and for `Real3` `-50.0`.
Description	This command sets the upper limits for the adjacent channel leakage power ratio measurement. `Real1` represents the 5 MHz lines in dBc and `Real2` represents the 10 MHz lines also in dBc. `Real3` represents the minimum power of the center frequency in dBm where this upper limits are valid.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFSP:ACLR:LIM:UPP -20.0,-25.0,-70.0`
	`:CALC:WCDM:RFSP:ACLR:LIM:UPP?`
	Returns `-20.0,-25.0,-70.0`.

:CALCulate:WCDMa:RFSPectrum:ACLR:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFSPectrum:ACLR:LIMit:LOWer:[DATA] <Real1>,<Real2>`
Parameters	`Realx` are floating point real numbers. The minimum value for the parameters is `-120.0`, the maximum is `0.0`, the resolution is `0.1`, the default value is for both values `-120.0`.
Description	This command sets the lower limits for the adjacent channel leakage power ratio measurement. The physical dimension of the number stated is dBc. `Real1` represents the `5 MHz` lines and `Real2` represents the `10 MHz` lines.
Query	There is a query form which returns the stored settings.

Example	:CALC:WCDM:RFSP:ACLR:LIM:LOW -60.0,-65.0 :CALC:WCDM:RFSP:ACLR:LIM:LOW? Returns -60.0,-65.0.

:CALCulate:WCDMa:RFSPectrum:SEM:LIMit[:FAIL]

Syntax	:CALCulate:WCDMa:RFSPectrum:SEM:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the adjacent channel leakage power ratio measurement exceeds the limits.
Example	:MEAS:WCDMA:RFSP:SEM:AVG? 20 :CALC:WCDM:RFSP:SEM:LIM? Returns 0 if the average spectrum line is within the template limits, otherwise 1.

:CALCulate:WCDMa:RFSPectrum:SEM:LIMit:STATe

Syntax	:CALCulate:WCDMa:RFSPectrum:SEM:LIMit:STATe <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: ON|OFF. Default is ON.
Description	This command switches the check of the measurement result values of the spectrum emission mask measurement against their user-definable limits either ON or OFF.
Query	There is no query form of this command available.
Example	:CALC:WCDM:RFSP:MSP:SEM:STAT ON Switches the limit check for the spectrum emission mask measurement on.

:CALCulate:WCDMa:RFSPectrum:SEM:LIMit:FREQuency

Syntax	:CALCulate:WCDMa:RFSPectrum:SEM:LIMit:FREQuency <Real1>,<Real2>,<Real3>,<Real4>,<Real5>,<Real6>,<Real7>,<Real8>,<Real9>,<Real10>,<Real11>,<Real12>
Parameters	Realx are floating point real numbers. The minimum value for the parameters 1 to 4 is -12.5, the maximum is -4.0, the minimum value for the parameters 5 to 8 is -3.5, the maximum is +3.5, the minimum value for the parameters 9 to 12 is +4.0, the maximum is +12.5, the resolution is 0.1. The default value is for Real1 -12.5, for Real2 -8.5, for Real3 -7.5, for Real4 -4.0, for Real5 -3.5, for Real6 -2.5, for Real7 +2.5, for Real8 +3.5, for Real9 +4.0, for Real10 +7.5, for Real11 +8.5, for Real12 +12.5.
Description	This command sets the frequency values for the limits for the spectrum emission mask measurement. Real1 to Real4 represent the frequency values of the left part of the spectrum emission mask (-12 MHz to -4 MHz) in MHz. Real5 to Real8 represent the frequency values of the middle part of the spectrum emission mask (-3.845 MHz to +3.845 MHz) in MHz. Real9 to Real12 represent the frequency values of the right part of the spectrum emission mask (+4 MHz to +12 MHz) in MHz.
Query	There is a query form which returns the stored settings.

Example

```
:CALC:WCDM:RFSP:SEM:LIM:FREQ -12,-8,-7,-5,-3,
-2,2,3,5,7,8,12
:CALC:WCDM:RFSP:SEM:LIM:FREQ?
```

Returns `-12.0,-8.0,-7.0,-5.0,-3.0,-2.0,2.0,3.0,5.0,7.0,8.0,12.0`.

:CALCulate:WCDMa:RFSPectrum:SEM:LIMit:UPPer:[DATA]

Syntax

```
:CALCulate:WCDMa:RFSPectrum:SEM:LIMit:UPPer:[DATA]
<Real1>,<Real2>,<Real3>,<Real4>,<Real5>,<Real6>,<Real7>
,<Real8>,<Real9>,<Real10>,<Real11>,<Real12>
```

Parameters

`Realx` are floating point real numbers. The minimum value for the parameters is `-120`, the maximum is `0`, the resolution is `1`, the default value is for `Real1 -49`, for `Real2 -49`, for `Real3 -39`, for `Real4 -35`, for `Real5 -50`, for `Real6 -35`, for `Real7 -35`, for `Real8 -50`, for `Real9 -35`, for `Real10 -39`, for `Real11 -49`, for `Real12 -49`.

Description

This command sets the upper limits for the spectrum emission mask measurement. `Real1` to `Real4` represents the upper limits of the left part of the spectrum emission mask (-12 MHz to -4 MHz) in dB. `Real5` to `Real8` represent the upper limits of the middle part of the spectrum emission mask (-3.845 MHz to +3.845 MHz) in dB. `Real9` to `Real12` represent the upper limits of the right part of the spectrum emission mask (+4 MHz to +12 MHz) in dB.

Query

There is a query form which returns the stored settings.

Example

```
:CALC:WCDM:RFSP:SEM:LIM:UPP -49,-49,-39,-35,-50,-35,
-35,-50,-35,-39,-49,-49
:CALC:WCDM:RFSP:SEM:LIM:UPP?
```

Returns `-49,-49,-39,-35,-50,-35,-35,-50,-35,-39,-49,-49`.

:CALCulate:WCDMa:RFSPectrum:SEM:LIMit:UPPer :ABSolute

Syntax

```
:CALCulate:WCDMa:RFSPectrum:SEM:LIMit:UPPer:ABSolute
<Real1>,<Real2>,<Real3>
```

Parameters

`Realx` are floating point real numbers. The minimum value for the parameters is `-120`, the maximum is `0`, the resolution is `1`, the default value is for `Real1 -55.843`, for `Real2 -71.072`, for `Real3 -55.843`.

Description

This command sets the minimum values for the upper limits for the spectrum emission mask measurement. `Real1` represents the minimum limit due to center channel power for the left part of the spectrum emission mask (-12 MHz to +4 MHz) in dB. `Real2` represents the minimum limit due to center channel power for the middle part of the spectrum emission mask (-3.845 MHz to +3.845 MHz) in dB. `Real3` represents the minimum limit due to center channel power for the right part of the spectrum emission mask (+4 MHz to +12 MHz) in dB.

Query

There is a query form which returns the stored settings.

Example

```
:CALC:WCDM:RFSP:SEM:LIM:UPP:ABS -55,-77,-55
:CALC:WCDM:RFSP:SEM:LIM:UPP:ABS?
```

Returns `-55,-77,-55`.

:CALCulate:WCDMa:RFSPectrum:SEM:VALue

Syntax	:CALCulate:WCDMa:RFSPectrum:SEM:VALue <real1>
Parameters	real1 is a floating point real number. The minimum value is -12,000,000.0, the maximum is 12,000,000.0. The minimum resolution possible is 0.1.
Description	There is only a query form of this command available.
Query	The query form of this command delivers an (interpolated) measurement result value of the modulation spectrum curve at the frequency specified with the real1 parameter. real1 is interpreted as an offset from the carrier (i.e. the transmission frequency). The string returned will contain one floating point real number. The physical dimension of the measurement result value returned is dBc. **Notes** – Please note that you need to have a spectrum measurement taken before this command will return any meaningful result. We recommend to use the :MEAS:WCDM:RFSP:SEM:AVG command to do so. – The :CALC:WCDM:RFSP:SEM:VAL command is not really a measurement - it is an interpolation of the spectrum curve for the frequency specified.
Example	:MEAS:WCDM:RFSP:SEM:AVG 20 :CALC:WCDM:RFSP:SEM:VAL? -50000 This command will interpolate a modulation spectrum measurement at an offset of 50 kHz to the carrier. The value returned in this example is: "-4.7".

:CALCulate:WCDMa:RFSPectrum:SEM:MAXPeak

Syntax	:CALCulate:WCDMa:RFSPectrum:SEM:MAXPeak?
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	The query form of this command delivers the frequency offset (in Hz) from the carrier and the maximum value of the measured modulation spectrum curve. The string returned will contain two floating point real numbers. The physical dimension of the measurement result values returned is for the first value Hz and for the second value dBc. **Notes** – Please note that you need to have a spectrum measurement taken before this command will return any meaningful result. We recommend to use the :MEAS:WCDM:RFSP:SEM:AVG command to do so. – The :CALC:WCDM:RFSP:MSP:MAXP command is not really a measurement - it is an readout of the spectrum curve.
Example	:MEAS:WCDM:RFSP:SEM:AVG 20 :CALC:WCDM:RFSP:SEM:MAXP? This command will read at the modulation spectrum measurement maximum the frequency offset to the carrier and the measurement result value. The values returned in this example is: "-67700.0,-5.2".

:CALCulate:WCDMa:RFRX:BER:LIMit[:FAIL]

Syntax	`:CALCulate:WCDMa:RFRX:BER:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is only a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the `:MEAS:WCDMA:ARR:RFRX:BER` measurement violated the defined limits.

Example	`:MEAS:WCDMA:ARR:RFRX:BER 20`
	`:CALC:WCDM:RFRX:BER:LIM?`
	The query returns `1` if any of the 20 measurement results was out of limits, or `0` if none of the results was out of limits.

:CALCulate:WCDMa:RFRX:BER:LIMit:STATe

Syntax	`:CALCulate:WCDMa:RFRX:BER:LIMit:STATe <PredefExp>`
Parameters	`PredefExp` can take on one of the following values: `ON|OFF`. The default is `ON`.
Description	This command switches the check of the measurement result values of the BER measurement against their user-definable limits either `ON` or `OFF`.
Query	There is a query form which returns the stored settings.
Example	`:CALC:WCDM:RFRX:BER:LIM:STAT ON` Switches the limit check for the BER measurement on.

:CALCulate:WCDMa:RFRX:BER:LIMit:UPPer:[DATA]

Syntax	`:CALCulate:WCDMa:RFRX:BER:LIMit:UPPer:[DATA] <real>`
Parameters	`real` is a floating point real number. Its minimum value is `0.0`, the maximum is `50.0`, the resolution is `0.1` and the default is `0.1`.
Description	Sets the upper limit for the mobile's Bit Error Rate. The physical dimension of the number is stated as a percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFRX:BER:LIM:UPP 2.5`
	`:CALC:WCDM:RFRX:BER:LIM:UPP?`
	Returns `2.5`.

:CALCulate:WCDMa:RFRX:BER:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFRX:BER:LIMit:LOWer:[DATA] <real>`
Parameters	`real` is a floating point real number. Its minimum value is `0.0`, the maximum is `1.0`, the resolution is `0.1` and the default is `0.0`.
Description	Sets the lower limit for the mobile's Bit Error Rate. The physical dimension of the number is stated as a percentage.
Query	There is a query form which returns the stored settings.

Example	:CALC:WCDM:RFRX:BER:LIM:LOW 0.0 :CALC:WCDM:RFRX:BER:LIM:LOW? Returns 0.0.

:CALCulate:WCDMa:RFRX:BLER:LIMit[:FAIL]

Syntax	:CALCulate:WCDMa:RFRX:BLER:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if the :MEAS:WCDMA:ARR:RFRX:BLER measurement violated the defined limits.
Example	:MEAS:WCDMA:ARR:RFRX:BLER 20 :CALC:WCDM:RFRX:BLER:LIM? The query returns 1 if any of the 20 measurement results was out of limits, or 0 if none of the results was out of limits.

:CALCulate:WCDMa:RFRX:BLER:LIMit:STATe

Syntax	:CALCulate:WCDMa:RFRX:BLER:LIMit:STATe <PredefExp>
Parameters	PredefExp can take on one of the following values: ON|OFF. The default is ON.
Description	This command switches the check of the measurement result values of the Bler measurement against their user-definable limits either ON or OFF.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFRX:BLER:LIM:STAT ON Switches the limit check for the Bler measurement on.

:CALCulate:WCDMa:RFRX:BLER:LIMit:UPPer:[DATA]

Syntax	:CALCulate:WCDMa:RFRX:BLER:LIMit:UPPer:[DATA] <real>
Parameters	real is a floating point real number. Its minimum value is 0.0, the maximum is 100.0, the resolution is 0.1 and the default is 0.1.
Description	Sets the upper limit for the mobile's Block Error Rate. The physical dimension of the number is stated as a percentage.
Query	There is a query form which returns the stored settings.
Example	:CALC:WCDM:RFRX:BLER:LIM:UPP 2.5 :CALC:WCDM:RFRX:BLER:LIM:UPP? Returns 2.5.

:CALCulate:WCDMa:RFRX:BER:LIMit:LOWer:[DATA]

Syntax	`:CALCulate:WCDMa:RFRX:BER:LIMit:LOWer:[DATA] <real>`
Parameters	`real` is a floating point real number. Its minimum value is `0.0`, the maximum is `1.0`, the resolution is `0.1` and the default is `0.0`.
Description	Sets the lower limit for the mobile's Block Error Rate. The physical dimension of the number is stated as a percentage.
Query	There is a query form which returns the stored settings.

Example	`:CALC:WCDM:RFRX:BLER:LIM:LOW 0.0`
	`:CALC:WCDM:RFRX:BLER:LIM:LOW?`
	Returns `0.0`.

RFGenerator subsystem

The RFG subsystem controls the accessible parameters of the RF generator.

Important notes:

- The RF generator can only be used if all communication systems have been switched off (and unloaded) before.
- The RF generator fuctionality of the 3100 will enable you to provide a base channel to allow the mobile under test to synchronize to the base station. However, as long as the RF generator is active, there will be **no call setup** and **no reaction to signaling**.
  Some of the data transmitted by the 3100 in the base channel can be set or altered using the SCPI commands described in section "CONFigure subsystem" on page 94.
- The RF generator and analyzer are enabled using the appropriate `:CONFigure:CSYStem` command.

:RFGenerator:GSM:STATe

Syntax	`:RFGenerator:GSM:STATe <PredefExp>`
Parameters	**`PredefExp`** is one of the following predefined expressions: **`ON`**|**`OFF`**. Default is **`OFF`**.
Description	Switches the RF generator on or off. While the Willtek 3100 is in RF generator mode, there is no signaling active. This means that the Willtek 3100 may be used as RF signal generator for all kinds of RF and GSM signals. While working as an RF generator, the Willtek 3100 does not respond to any messages sent by the mobile nor does the test set expect the mobile under test to react in any way.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**`:RFGenerator:GSM:STATe ON`** **`:RFG:GSM:STAT?`** Value returned: **`"ON"`**.

:RFGenerator:GSM:LEVel

Syntax	:RFGenerator:GSM:LEVel <real1>
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **-120.0**, the maximum value is **-10.0**. The minimum resolution for **real1** is **0.1**. The default value for **real1** is **-60.0**.
Description	This command sets the RF output power level of the Willtek 3100. The value specified for **real1** is the power output level in dBm.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.
Example	**:RFGenerator:GSM:LEVel -50.5** **:RFG:GSM:LEV?** Value returned: **"-50.5"**.

:RFGenerator:GSM:FREQuency

Syntax	:RFGenerator:GSM:FREQuency <real1>
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **800,000,000**, the maximum value is **2,000,000,000**. The minimum resolution for **real1** is **10**. The default value for **real1** is **800,000,000**.
Description	This command sets the RF generator's frequency. The value specified for **real1** is the output frequency in Hz. **Note:** For more details on the GSM frequency ranges, please refer to section Basic Specifications of GSM Bands.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.
Example	**:RFGenerator:GSM:FREQuency 880200010** **:RFG:GSM:FREQ?** Value returned: **"880200010"**.

:RFGenerator:GSM:MODulation

Syntax	:RFGenerator:GSM:MODulation <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **NO|GMSK|AM**. Default is **NO**.
Description	This command selects the type of modulation to be used on the RF output signal. **NO** means that the output signal will **not** be modulated, while **GMSK** means that the output signal will be GMSK (**G**aussian **M**inimum **S**hift **K**eying) modulated. Another alternative is **A**mplitude **M**odulation. Notes – As soon as **GMSK** is selected, the Willtek 3100 will provide a GSM-like RF output signal. A number of GSM-specific parameters (like burst transmission or the type of training sequence to be integrated into the bursts) may be selected with the commands described below. – AM is available only if the AM Signal Generator Option is installed.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.

Example	**:RFGenerator:GSM:MODulation GMSK** **:RFG:GSM:MOD?** Value returned: **"GMSK"**.

:RFGenerator:GSM:MODulation:AM:MFRequency

Syntax	:RFGenerator:GSM:MODulation:AM:MFRequency <real1>
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **1,000**, the maximum value is **50,000**. The resolution for **real1** is **1000**. The default value for **real1** is **10,000**.
Description	This command sets the modulating frequency for the AM modulated signal. **Note:** AM is available only if the AM Signal Generator Option is installed.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.
Example	**:RFGenerator:GSM:MOD:AM:MFR 2000** **RFG:GSM:MOD:AM:MFR?** Value returned: **"2000"**.

:RFGenerator:GSM:MODulation:AM:MDEPth

Syntax	:RFGenerator:GSM:MODulation:AM:MDEPth <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **50**, the maximum value is **90**. The default value is **83**.
Description	This parameter describes the AM modulation depth in %. **Note:** AM is available only if the AM Signal Generator Option is installed.
Query	The query form of this command returns the current setting. A query will return a string containing one integer.
Example	**:RFGenerator:GSM:MOD:AM:MDEP 50** **:MOD:AM:MDEP?** Value returned: **"50"**.

:RFGenerator:GSM:MODulation:POLarity

Syntax	:RFGenerator:GSM:MODulation:POLarity <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **NORMal|INVert**. Default is **NORMal**.
Description	Sets the modulation polarity for the GMSK coded RF output signals. **NORMal** means standard modulation polarity while **INVert** will invert the modulation polarity.
Query	The query form of this command returns the current setting. A query will return a string, containing the short-form version of one of the predefined expressions explained above.
Example	**:RFG:GSM:MODulation:POLarity INVert** **:RFG:MOD:POL?** Value returned: **"INV"**.

:RFG:GSM:MODulation:TSEQuence:STATe

Syntax	:RFG:GSM:MODulation:TSEQuence:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	This command switches the use of a training sequence either on or off. Notes – The type of training sequence can be set using the command described below. – More details regarding the training sequence can be found in section Timeslots and Bursts.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**:RFG:GSM:MODulation:TSEQuence:STATe ON** **:RFG:MOD:TSEQ:STAT?** Value returned: **"ON"**.

:RFG:GSM:MODulation:TSEQuence[:DATA]

Syntax	:RFG:GSM:MODulation:TSEQuence[:DATA] <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **7**. The default value is **0**. **Note:** This command only selects the type of training sequence. To switch the use of a training sequence on or off, use the command described above.
Description	This command selects the type of training sequence to be used. More details regarding the training sequence can be found in section Timeslots and Bursts.
Query	The query form of this command returns the current setting. A query will return a string, containing one integer.
Example	**:RFG:GSM:MODulation:TSEQuence:DATA 3** **:RFG:MOD:TSEQ?** Value returned: **"3"**.

:RFG:GSM:MODulation:BITPattern

Syntax	:RFG:GSM:MODulation:BITPattern <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **PRBS9|PRBS15|PRBS23|ALLZero|ALLOne|** **ONEZero|DOUBleonezer|FOURonezero|** **EIGHtonezero**. Default is **PRBS9**.

Description	This command selects the bit sequence to be transmitted by the RF generator in GMSK mode on the 'encrypted data' positions (see section Timeslots and Bursts for reference). When **:RFG[:GSM]:MOD:TSEQ:STAT OFF** has been selected, all 148 'useful' bits of a timeslot will be filled with the bit pattern specified with this command. **PRBS9** stands for a pseudorandom bit sequence. The length of this sequence will be 511 bits ((2 to the power of 9)-1). **PRBS15** is a pseudorandom bit sequence with a length of 32767 bits. **PRBS23** is a pseudorandom bit sequence as well. The length is 8388607 bits. **ALLZero** means that all bits transmitted are 'zeros' (**000...**). In case, **ALLOne** is selected, all bits transmitted will be set to 'ones' (**111...**). **ONEZero** stands for a bit sequence starting with a 'one' and continuing with the inverted value of the current bit (**10101...**). **DOUBleonezero** will set the bits transmitted to a sequence of double 'ones' and double 'zeros' (**11001100...**). **FOURonezero** means a bit sequence starting with four 'ones' and continuing with the inverted value of the current four bits (**111100001111...**), while **EIGHtonezero** means a bit sequence starting with eight 'ones' and continuing with the inverted value of the current eight bits (**111111110000000011111111...**).
Query	The query form of this command will return the current setting. The string delivered back will contain one of short-form versions of the predefined expressions explained above.
Example	**:RFG:GSM:MODulation:BITPattern PRBS15** **:RFG:MOD:BITP?** Value returned: **"PRBS15"**.

:RFG:GSM:MODulation:DIFFbitcod

Syntax	:RFG:GSM:MODulation:DIFFbitcod <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON | OFF**. Default is **ON**.
Description	This command switches differential bit coding either on or off.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**:RFGenerator:GSM:MODulation:DIFFbitcod ON** **:RFG:MOD:DIFF?** Value returned: **"ON"**.

:RFGenerator:GSM:BURSt:TSLot

Syntax	:RFGenerator:GSM:BURSt:TSLot <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **7**. The default value is **0**.
Description	This command selects the time slot during which the Willtek 3100 will transmit its burst. **Note:** With the help of command **RFG:STYP** you may select whether the Willtek 3100 transmits a continuous wave or bursts. This command is described below.
Query	The query form of this command returns the current setting. A query will return a string, containing one integer.

Example	**:RFGenerator:GSM:BURSt:TSLot 3** **:RFG:BURS:TSL?** Value returned: **"3"**.

:RFGenerator:GSM:STYPe

Syntax	:RFGenerator:GSM:STYPe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **BURSt|CWAVe**. Default is **BURSt**.
Description	This command switches between burst transmission (**BURSt**) or continuous signal transmission without bursts (**CWAVe**). **CWAVe** means that the Willtek 3100 will generate a coded GSM signal **in all** timeslots according to the settings above. When set to **BURSt**, the Willtek 3100 will generate only one burst per TDMA frame. This means that there will be no RF transmission during the remaining 7 timeslots.
Query	The query form of this command returns the current setting. A query will return a string, containing the short-form version of one of the predefined expressions explained above.
Example	**:RFGenerator:GSM:STYPe BURSt** **:RFG:STYP?** Value returned: **"BURS"**.

:RFGenerator:GSM:MSLot:LEVel[:DATA]

Syntax	:RFGenerator:GSM:MSLot:LEVel[:DATA] <real1>,<real2>,<real3>,<real4>,<real5> ,<real6>,<real7>,<real8>
Parameters	**real1...real8** are eight floating point real numbers. The minimum value for each **realx** is **-120.0**, the maximum value is **-10.0**. The resolution of each **realx** is **0.1**. The default value for each **realx** is **-60.0**. **Note:** All eight values **must** be specified.
Description	This command configures individual RF output power levels for the eight DL timeslots. Every value represents the RF output power level of the Willtek 3100 in dBm during the related timeslot. **Note:** This command **only prepares** the use of individual RF power levels. To activate them, use the **RFG:MSL:LEV:MODE** command.
Query	The query form of this command will return the current setting. Always eight floating point real numbers will be returned in a string.
Example	**:RFGenerator:GSM:MSLot:LEVel:DATA -80.0,-70.0,-60.0, -50.0,-40.0, -30.0,-20.0,-10.0** **:RFG:MSL:LEV?** Values returned: **"-80.0,-70.0,-60.0,-50.0,-40.0, -30.0, -20.0,-10.0"**

:RFGenerator:GSM:MSLot:LEVel:MODE

Syntax	:RFGenerator:GSM:MSLot:LEVel:MODE <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **INDividual|STANdard**. Default is **INDividual**.

Description	Switches the use of individual RF output power levels on or off. **INDividual** means that the individual RF output power levels, specified with the **:RFG:MSL:LEV** command will be used. **STANdard** means that the individual RF power levels will be ignored and that the standard RF output level of the Willtek 3100 (as set with the **:RFG:LEV** command) will be used. **Note:** For a detailed explanation of the individual power levels in co-junction with the multislot measurement capabilities of the Willtek 3100 (HSCSD), please refer to section Frequently Asked Questions.
Query	The query form of this command will return the current setting. The string delivered back will contain the short-form version of one of the predefined expressions explained above.
Example	**:RFG:GSM:MSLot:LEVel:MODE INDividual** **:RFG:MSL:LEV:MODE?** Value returned: **"IND"**.

:RFGenerator:WCDMa:STATe

Syntax	:RFGenerator:WCDMa:STATe <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: ON|OFF. Default is OFF.
Description	Switches the RF generator ON or OFF. While the Willtek 3100 is in RF generator mode, there is no signalling active. This means that the Willtek 3100 may be used as RF signal generator for all kinds of RF and WCDMA signals. While working as a RF generator, the Willtek 3100 does not respond to any messages sent by the mobile nor does the test set expect the mobile under test to react in any way.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	:RFG:WCDMA:STAT ON :RFG:WCDMA:STAT? Value returned in this example "ON".

:RFGenerator:WCDMa:LEVel

Syntax	:RFGenerator:WCDMa:LEVel <real1>
Parameters	real1 is a floating point real number. The minimum value for real1 is -120.0, the maximum value is -20.0. The minimum resolution for real1 is 0.1. The default value for real1 is -60.0.
Description	This command sets the RF output power level of the Willtek 3100. The value specified for real1 is the power output level in dBm.
Query	The query form of this command will return the current setting. The string delivered will contain one floating point real number.
Example	:RFG:WCDMA:LEV -50.5 :RFG:WCDMA:LEV? Value returned in this example: -50.5.

:RFGenerator:WCDMa:FREQuency

Syntax	:RFGenerator:WCDMa:FREQuency <real1>
Parameters	real1 is a floating point real number. The minimum value for real1 is 430,000,000, the maximum value is 2,300,000,000. The minimum resolution for real1 is 10. The default value for real1 is 800,000,000.
Description	This command sets the RF generator's frequency. The value specified for real1 is the output frequency in Hz. **Note:** For more details on the WCDMA frequency ranges, please refer to section Basic Specifications of WCDMA Bands.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.
Example	:RFG:WCDMA:FREQ 880200010 :RFG:WCDMA:FREQ? Value returned in this example: 880200010.

:RFGenerator:WCDMa:UARFcn

Syntax	:RFGenerator:WCDMa:UARFcn <int1>
Parameters	int1 is an integer number. The minimum value is 0, the maximum value 10838. The initial default is 10700.
Description	This command will set the Downlink channel number of the 3100.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:RFG:WCDMA:UARF 2010 :RFG:WCDMA:UARF? Value returned in this example: 2010.

:RFGenerator:WCDMa:FOFFset

Syntax	:RFGenerator:WCDMa:FOFFset <int1>
Parameters	int1 is an integer number. The minimum value is −200000, the maximum value +200000. The initial default is 0.
Description	This command will set the Downlink frequency offset of the 3100.
Query	The query form of this command will return the current setting. The string delivered will contain one integer.
Example	:RFG:WCDMA:FOFF 30000 :RFG:WCDMA:FOFF? Value returned in this example: 30000.

:RFGenerator:WCDMa:BAND?

Syntax	`:RFGenerator:WCDMa:BAND?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	This query determines the current Signal Band (based on the current UARFCN).
Example	`:RFG:WCDMA:BAND?` Value returned in this example: `2`.

:RFGenerator:WCDMa:MODulation

Syntax	`:RFGenerator:WCDMa:MODulation <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `NONE|FM|WCDMA`. Default is `NONE`.
Description	This command selects the type of modulation to be used on the RF output signal. `NONE` means that the output signal will not be modulated, while `FM` means that the output signal will be FM modulated and `WCDMa` means that the Willtek 3100 will provide a WCDMA-like RF output signal.
Query	The query form of this command returns the current setting. A query will return a string containing one of the predefined expressions explained above.

Example	`:RFG:WCDMA:MOD WCDMa`
	`:RFG:WCDMA:MOD?`
	Value returned in this example `"WCDM"`.

:RFGenerator:WCDMa:MODulation:FM:MFRequency

Syntax	`:RFGenerator:WCDMa:MODulation:FM:MFRequency <real1>`
Parameters	`real1` is a floating point real number. The minimum value for `real1` is `1,000`, the maximum value is `100,000`. The minimum resolution for `real1` is `10`. The default value for `real1` is `50,000`.
Description	This command sets the modulation frequency for the generated FM signal.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:RFG:WCDMA:MOD:FM:MFR 30000`
	`:RFG:WCDMA:MOD:FM:MFR?`
	Value returned in this example: `30000`.

:RFGenerator:WCDMa:MODulation:FM:FDEViation

Syntax	`:RFGenerator:WCDMa:MODulation:FM:FDEViation <real1>`
Parameters	`real1` is a floating point real number. The minimum value for `real1` is `250,000`, the maximum value is `1,000,000`. The minimum resolution for `real1` is `10`. The default value for `real1` is `500,000`.

Description	This command sets the Modulation Frequency for the generated FM signal.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:RFG:WCDMA:MOD:FM:FDEV 300000`
	`:RFG:WCDMA:MOD:FM:FDEV?`
	Value returned in this example: `300000`.

:RFGenerator:WCDMa:MODulation:WCDMa:BITPattern

Syntax	`:RFGenerator:WCDMa:MODulation:WCDMa:BITPattern <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `PRBS9|PRBS15|PRBS23|ALLZero|ALLOne|ONEZero|ZEROone`. The default is `PRBS9`.
Description	This command sets the bit pattern for the WCDMA signal.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.

Example	`:RFG:WCDMA:MOD:WCDM:BITP ONEZero`
	`:RFG:WCDMA:MOD:WCDM:BITP?`
	Value returned in this example `"ONEZero"`.

:RFGenerator:WCDMa:MODulation:WCDMa:SCODe

Syntax	`:RFGenerator:WCDMa:MODulation:WCDMa:SCODe <int1>`
Parameters	`int1` is an integer number. The minimum value is `0`, the maximum value `15`. The initial default is `0`.
Description	This command sets the DPCH (secondary) scrambling code for the WCDMA signal.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:RFG:WCDMA:MOD:WCDM:SCOD 10`
	`:RFG:WCDMA:MOD:WCDM:SCOD?`
	Value returned in this example: `10`.

:RFGenerator:WCDMa:MODulation:WCDMa:CCODe

Syntax	`:RFGenerator:WCDMa:MODulation:WCDMa:CCODe <int1>`
Parameters	`int1` is an integer number. The minimum value is `3`, the maximum value `511`. The initial default is `3`.
Description	This command sets the channelization code for the WCDMA signal.
Query	The query form of this command will return the current setting. The string delivered will contain one integer.

Example	`:RFG:WCDMA:MOD:WCDM:CCOD 10`
	`:RFG:WCDMA:MOD:WCDM:CCOD?`
	Value returned in this example: `10`.

:RFGenerator:WCDMa:MODulation:WCDMa:SFORmat

Syntax	`:RFGenerator:WCDMa:MODulation:WCDMa:SFORmat <int1>`
Parameters	`int1` is an integer number. The minimum value is `0`, the maximum value `16`. The initial default is `11`.
Description	This command sets the slot format for the WCDMA signal.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:RFG:WCDMA:MOD:WCDM:SFOR 10`
	`:RFG:WCDMA:MOD:WCDM:SFOR?`
	Value returned in this example: `10`.

:RFGenerator:WCDMa:MODulation:WCDMa:SFACtor?

Syntax	`:RFGenerator:WCDMa:MODulation:WCDMa:SFACtor?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	This query determines the spreading factor for the WCDMA signal.
Example	`:RFG:WCDMA:MOD:WCDM:SFAC?` Value returned in this example: `20`.

:RFGenerator:WCDMa:MODulation:WCDMa:DSTAte

Syntax	`:RFGenerator:WCDMa:MODulation:WCDMa:DSTAte <PredefExp>`
Parameters	PredefExp is one of the following predefined expressions: `ON|OFF`. Default is `ON`.
Description	This command switches the DPCH code channel `ON` or `OFF`.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.

Example	`:RFG:WCDMA:MOD:WCDM:DSTA ON`
	`:RFG:WCDMA:MOD:WCDM:DSTA?`
	Value returned in this example "`ON`".

:RFGenerator:WCDMa:TXCorrection

Syntax	`:RFGenerator:WCDMa:TXCorrection`
Parameters	There are no parameters.
Description	This command sends a message to start the TX correction procedure.
Query	There is no query form of this command.
Example	`:RFG:WCDMA:TXC`

:RFG:WCDMa:TXC:CAL

Syntax	`:RFG:WCDMa:TXC:CAL?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers a boolean number in a string which indicates if a calibration command should be issued.
Example	`:RFG:WCDMA:TXC:CAL?` The query returns `1` if a `RFG:WCDMA:TXC` command needs to be issued, or `0` if ok.

RFANalyser subsystem

The RFAN subsystem controls the accessible parameters of the RF analyzer.

:RFANalyser:GSM:FREQuency

Syntax	:RFANalyser:GSM:FREQuency <real1>
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **800,000,000**, the maximum value is **2,000,000,000**. The minimum resolution for **real1** is **1**. The default value for **real1** is **800,000,000**.
Description	This command sets the RF analyzer's center frequency. The value specified for **real1** is the center frequency in Hz. Notes – The frequency of the Willtek 3100 internal synthesizer can be changed in steps of 10 Hz. – For further details on the RF analyzer, please refer to section **3.1 Generator/Analyzer**.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.
Example	**:RFANalyser:GSM:FREQuency 880200000** **:RFAN:FREQ?** Value returned: **"880200000"**.

:RFANalyser:GSM:MODulation

Syntax	:RFANalyser:GSM:MODulation <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **GMSK|UNMod**. Default is **GMSK**.
Description	Sets the modulation type to be expected by the RF analyzer. By nature, measurement results differ depending on the expected type of modulation.
Query	Returns the current setting, which is either **GMSK** or **UNM**.
Example	**:RFANalyser:GSM:MODulation GMSK** **::RFANalyser:GSM:MODulation?** Value returned: **"GMSK"**.

:RFANalyser:WCDMa:FREQuency

Syntax	:RFANalyser:WCDMa:FREQuency <Real1>
Parameters	Real1 is a floating point real number. The minimum value for Real1 is 430,000,000, the maximum value is 2,300,000,000. The minimum resolution for real1 is 10. The default value is 800,000,000.
Description	This command sets the RF analyzer's center frequency. The value specified is the center frequency in Hz. The frequency of the internal synthesizer can be changed in steps of 10 Hz.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.

Example	:RFAN:WCDM:FREQ 1950000000 :RFAN:WCDM:FREQ? Value returned in this example: 1950000000.

:RFANalyser:WCDMa:UARFcn

Syntax	:RFANalyser:WCDMa:UARFcn <int1>
Parameters	int1 is an integer number. The minimum value is 0, the maximum value 9888. The initial default is 9750.
Description	This command will set the Uplink channel number of the 3100.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:RFAN:WCDM:UARF 6510 :RFAN:WCDM:UARF? Value returned in this example: 6510.

:RFANalyser:WCDMa:FOFFset

Syntax	:RFANalyser:WCDMa:FOFFset <int1>
Parameters	int1 is an integer number. The minimum value is −200000, the maximum value +200000. The initial default is 0.
Description	This command will set the Uplink frequency offset of the 3100.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:RFAN:WCDM:FOFF 30000 :RFAN:WCDM:FOFF? Value returned in this example: 30000.

:RFANalyser:WCDMa:SCODe

Syntax	:RFANalyser:WCDMa:SCODe <string1>
Parameters	string1 is a string of up to 6 characters.
Description	This command sets the scrambling code used by the RF analyzer for WCDMA measurements.
Query	The query form of this command will return the current setting of the Willtek 3100.
Example	:RFAN:WCDM:SCOD AABBCD :RFAN:WCDM:SCOD? Value returned in this example: "AABBCD".

:RFANalyser:WCDMa:SRATe

Syntax	:RFANalyser:WCDMa:SRATe <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: SR60|SR15|SR30|SR120|SR240|SR480|SR960. Default is SR60.
Description	This command will set the symbol rate of the 3100 Analyzer.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	:RFAN:WCDM:SRAT SR15 :RFAN:WCDM:SRAT? Value returned in this example "SR15".

:RFANalyser:WCDMa:SRATe

Syntax	:RFANalyser:WCDMa:SRATe <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: SR60|SR15|SR30|SR120|SR240|SR480|SR960. Default is SR60.
Description	This command will set the symbol rate of the 3100 Analyzer.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	:RFAN:WCDM:SRAT SR15 :RFAN:WCDM:SRAT? Value returned in this example "SR15".

:RFANalyser:WCDMa:CCODe

Syntax	:RFANalyser:WCDMa:C0CODe <int1>
Parameters	int1 is an integer number. The minimum value is 1, the maximum value 256. The initial default is 16.
Description	This command will set the channel code of the 3100 Analyzer.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:RFAN:WCDM:CCOD 10 :RFAN:WCDM:CCOD? Value returned in this example: 10.

:RFANalyser:WCDMa:MODulation:WCDMa:SFORmat

Syntax	:RFANalyser:WCDMa:MODulation:WCDMa:SFORmat <int1>
Parameters	int1 is an integer number. The minimum value is 0, the maximum value 5. The initial default is 0.

Description	This command will set the slot format of the 3100 Analyzer.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:RFAN:WCDM:SFOR 2`
	`:RFAN:WCDM:SFOR?`
	Value returned in this example: `2`.

:RFANalyser:WCDMa:NSTaircase:FREQuency

Syntax	`:RFANalyser:WCDMa:NSTaircase:FREQuency <Real1>`
Parameters	`Real1` is a floating point real number. The minimum value for `Real1` is `430,000,000`, the maximum value is `2,300,000,000`. The minimum resolution for `real1` is `10`. The default value is `1,950,300,000`.
Description	This command will set the Uplink frequency of the 3100 in Hz for Staircase Measurements. The frequency of can be changed in steps of 10 Hz.
Query	The query form of this command will return the current setting. The string delivered will contain one floating point real number.

Example	`:RFAN:WCDM:NST:FREQ 1950000000`
	`:RFAN:WCDM:NST:FREQ?`
	Value returned in this example: `1950000000`.

:RFANalyser:WCDMa:NSTaircase:UARFcn

Syntax	`:RFANalyser:WCDMa:NSTaircase:UARFcn <int1>`
Parameters	`int1` is an integer number. The minimum value is `0`, the maximum value `9888`. The initial default is `9750`.
Description	This command will set the Uplink channel number of the 3100 for Staircase measurements.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:RFAN:WCDM:NST:UARF 7510`
	`:RFAN:WCDM:NST:UARF?`
	Value returned in this example: `7510`.

:RFANalyser:WCDMa:ZSPan:REFLevel

Syntax	`:RFANalyser:WCDMa:ZSPan:REFLevel <real1>`
Parameters	`real1` is a floating point real number. The minimum value for `real1` is `-23.0`, the maximum value is `36.0`. The minimum resolution for `real1` is `0.1`. The default value for `real1` is `0.0`.
Description	This command sets the reference level of the zero span Analyzer (upper limit). The value specified for real1 is in dBm.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.

Example	:RFAN:WCDM:ZSP:REFL -10.5 :RFAN:WCDM:ZSP:REFL? Value returned in this example: -10.5.

:RFANalyser:WCDMa:ZSPan:TRGLevel

Syntax	:RFANalyser:WCDMa:ZSPan:TRGLevel <real1>
Parameters	real1 is a floating point real number. The minimum value for real1 is -60.0, the maximum value is 36.0. The minimum resolution for real1 is 0.1. The default value for real1 is 0.0.
Description	This command sets the trigger level of the zero span Analyzer. The value specified for real1 is in dBm.
Query	The query form of this command will return the current setting. The string delivered will contain one floating point real number.
Example	:RFAN:WCDM:ZSP:TRGL -10.5 :RFAN:WCDM:ZSP:TRGL? Value returned in this example: -10.5.

:RFANalyser:WCDMa:ZSPan:RES

Syntax	:RFANalyser:WCDMa:ZSPan:RES <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: RES30|RES100|RES46848. Default is RES46848.
Description	This command will set the resolution bandwidth for zero span measurements.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	:RFAN:WCDM:ZSP:RES RES100 :RFAN:WCDM:ZSP:RES? Value returned in this example "RES100".

:RFANalyser:WCDMa:ZSPan:RERFoints

Syntax	:RFANalyser:WCDMa:ZSPan:REFPoints <real1><real2>...<realn>
Parameters	real1 is an integer number. The minimum value is -17.000 sec, the maximum value 50.000 sec. The initial default is 0.000 sec. real2 to realn are real numbers. The minimum value is 0.000 sec, the maximum value 17.000 sec. The initial default is 5.000 sec. Minimum are 2 parameters RP0, RP1.

Description	The reference points `RP0..RPn` define timing intervals for measuring and not measuring. The timing interval `RP0` will be not measured, after that alternating measure and non-measure intervals follow. Every timing interval `RPn`; `RPn+1` can be between `0.0 sec` and `17.000 sec`. Max. length of sequence due to ACQ-RAM: `17.000 sec` (without consideration of the first `RP0`-interval). `RP0`: `-17000 sec` to `+50000 sec` => `RP0` correspond to "PRE-TRIGGER" in zerospan graphic display. `RP1..RPn`: `0.0 sec` to `17.000 sec` => `RP1` correspond to "Sweep-Time" in zerospan graphic display. Sum of `RP1` to `RPn` must be =< `17.000 sec`.
Query	The query form of this command will return the current settings.
Example	`:RFAN:WCDM:ZSP:REFP 0,1.000,2.000,3.000,4.000,5.000,6.000,7.000,8.000, 9.000,10.000,11.000,12.000,13.000,14.000,15.000,16.000, 17.000`

RFSPectrum subsystem

The RFSP subsystem controls the accessible parameters of the RF modulation spectrum analyzer.

`:RFSPectrum[:GSM]:MSPectrum:SPAN`

Syntax	`:RFSPectrum[:GSM]:MSPectrum:SPAN <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `SP400|SP1000|SP3600`. Default is `SP400`.
Description	Sets the span (i.e. the spectral bandwidth) of the modulation spectrum analyzer. `SP400` will set a span of 400 kHz, `SP1000` will set a span of 1 MHz. `SP3600` will set a span of 3.6 MHz. **Note:** More details regarding the modulation spectrum analyzer can be found in section **Spectrum Measurements**.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	`:RFSP:MSP:SPAN SP1000` `:RFSP:MSP:SPAN?` Value returned in this example "`SP1000`".

`:RFSPectrum[:GSM]:MSPectrum:RESolution`

Syntax	`:RFSPectrum[:GSM]:MSPectrum:RESolution <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `RES5|RES10|RES30`. Default is `RES2`.

Description	Sets the spectral resolution of the modulation spectrum analyzer. `RES5` stands for a resolution of 5 kHz. This resolution may only be chosen in conjunction with the 400 kHz spectral bandwidth (span) as described above. `RES5` will set a resolution of 5 kHz. This resolution may only be chosen in conjunction with the 400 kHz and 1 MHz spectral bandwidth (span) as described above. `RES10` means a resolution of 10 kHz, while `RES30` will set a resolution of 30 kHz. This resolution may only be chosen in conjunction with the 3.6 MHz spectral bandwidth (span) as described above. **Notes** – With the span set to 400 kHz, only the following spectral resolutions are available: 2 kHz (`RES2`), 5 kHz (`RES5`) and 10 kHz (`RES10`). – With the span set to 1 MHz, only the following spectral resolutions are available: 5 kHz (`RES5`) and 10 kHz (`RES10`). – With the span set to 3.6 MHz, only the following spectral resolutions are available: 10 kHz (`RES10`) and 30 kHz (`RES30`). More details regarding the modulation spectrum analyzer can be found in section **Spectrum Measurements** Please note that illegal combinations of span and resolution may be accepted by the Willtek 3100 and might lead to undefined measurement results.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.

Example	`:RFSP:MSP:RES RES10`
	`:RFSP:MSP:RES?`
	Value returned in this example `"RES10"`.

:RFSPectrum:WCDMa:MSPectrum:SPAN

Syntax	`:RFSPectrum:WCDMa:MSPectrum:SPAN <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `SP10|SP6`. The default is `SP10`.
Description	Sets the span (i.e. the spectral bandwidth) of the modulation spectrum analyzer. `SP6` will set a span of 6 MHz (+/- 3 MHz), `SP10` will set a span of 10 MHz (+/- 5 MHz).
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.

Example	`:RFSP:WCDMA:MSP:SPAN SP10`
	`:RFSP:WCDMA:MSP:SPAN?`
	Value returned in this example `SP10`.

:RFSPectrum:WCDMA:MSPectrum:RESolution

Syntax	`:RFSPectrum:WCDMA:MSPectrum:RESolution <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `RES15|RES30`. The default is `RES15`.
Description	Sets the spectral resolution of the modulation spectrum analyzer. `RES30` will set a resolution of 30 kHz. `RES15` will set a resolution of 15 kHz.

Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.

Example	`:RFSPectrum:WCDMA:MSPectrum:RESolution RES15`
	`:RFSP:WCDMA:MSP:RES?`
	Value returned in this example `"RES15"`.

:RFSPectrum:WCDMa:MSPectrum:SEM:SPAN

Syntax	`:RFSPectrum:WCDMa:MSPectrum:SEM:SPAN <PredefExp>`
Parameters	`PredefExp` is the following predefined expression: `SP12`. The default is `SP12`.
Description	Sets the span (i.e. the spectral bandwidth) of the spectrum emission mask spectrum analyzer. `SP12` will set a span of 12 MHz.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.

Example	`:RFSP:WCDMA:MSP:SEM:SPAN SP12`
	`:RFSP:WCDMA:MSP:SEM:SPAN?`
	Value returned in this example `SP12`.

:RFSPectrum:WCDMa:MSPectrum:SEM:RESolution

Syntax	`:RFSPectrum:WCDMa:MSPectrum:SEM:RESolution <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `RES30`. The default is `RES30`.
Description	Sets the spectral resolution of the spectrum emission mask spectrum analyzer. `RES30` will set a resolution of 30 kHz.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.

Example	`:RFSP:WCDMA:MSP:SEM:RES RES30`
	`:RFSP:WCDMA:MSP:SEM:RES?`
	Value returned in this example `RES30`.

AFGenerator subsystem

The AFG subsystem controls the accessible parameters of the audio generator. Please note that all commands of this subsystem require the Audio Option to be installed on your 3100.

:AFGenerator:STATe

Syntax	:AFGenerator:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **OFF**.
Description	Switches the AF (audio frequency) generator on or off. **Note:** This command requires the Audio Option to be installed on your Willtek 3100.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**:AFGenerator:STATe ON** **:AFG:STAT?** Value returned: **"ON"**.

:AFGenerator:MODE

Syntax	:AFGenerator:MODE <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **SINGle|REPLay**. Default is **SINGle**.
Description	This command sets the AF generator either to **SINGle** tone generator mode or to wave file **REPLay** generator.
Query	The query form of this command returns the current setting. A query will return a string containing the short-form version of one of the predefined expressions explained above.
Example	**:AFGenerator:MODE REPLay** **:AFG:MODE?** Value returned: **"REPL"**.

:AFGenerator:OUTPut

Syntax	:AFGenerator:OUTPut <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: AF|CODec. Default is AF.
Description	This command routes the signal of the AF generator. AF means that the generator's signal will be available at the AF out connector, located on the front panel of the Willtek 3100. CODec means that the signal created by the AF generator will be output to the (internal) codec of the Willtek 3100. **Note:** RF cannot be signal source and generator destination the same time. Please check the current setting of the AFG:OUTP command before issuing the AFAN:INP COD command and vice versa to avoid measurement conflicts.

Query	The query form of this command returns the current setting. A query will return a string, containing the short-form version of one of the predefined expressions explained above.
Example	`:AFGenerator:OUTPut CODec` `:AFG:OUTP?` Value returned: `"COD"`.

:AFGenerator:LEVel[:RELative]:AF

Syntax	`:AFGenerator:LEVel[:RELative]:AF <real1>`
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **0.000**, the maximum value is **4.000**. The resolution for **real1** is **0.001**. The default value for **real1** is **0.000**.
Description	This command sets the output voltage of the signals, available at the AF out connector, located on the front panel of the Willtek 3100. The value specified for **real1** is the **peak voltage** of the AF signal. **Notes** – This command will only have an effect when **AF** has been selected as the output with the **AFG:OUTP** command explained above. – The maximum output voltage for sinusoidal signals is 4 V (rms) (that is equivalent to 5.6 V peak).
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.
Example	**:AFGenerator:LEVel[:RELative]:AF 2.5** **:AFG:LEV:AF?** Value returned in this example: **"2.5"**.

:AFGenerator:LEVel[:RELative]:RF

Syntax	`:AFGenerator:LEVel[:RELative]:RF <real1>`
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **-55.0**, the maximum value is **0.0**. The minimum resolution for **real1** is **0.1**. The default value for **real1** is **-6.0**.
Description	This command sets the 'modulation depth' of the audio signal in case it is output through the codec and thus used to modulate the RF output of the Willtek 3100. The physical dimension of **real1** is dBFS (dB full scale). **Notes** – A setting of 0 dB means that the audio signal will use the full input range of the codec, while -6 dB for instance mean that only half the input range will be used. – This command will only have an effect when **CODec** has been selected as the output with the **AFG:OUTP** command explained above.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.
Example	**:AFGenerator:LEVel[:RELative]:RF -12.0** **:AFG:LEV:RF?** Value returned: **"-12.0"**.

:AFGenerator:SINGle:FREQuency

Syntax	`:AFGenerator:SINGle:FREQuency <real1>`

Parameters	**real1** is a floating point real number. The minimum value for **real1** is **20.0**, the maximum value is **20000.0**. The minimum resolution for **real1** is **0.1**. The default value for **real1** is **1000.0**.
Description	This command sets the frequency of the AF generator as long as it is run in single-tone generation mode. The physical dimension of **real1** is Hertz. **Note:** Please refer to the description of the **:AFG:MODE** command for further details regarding the single-tone generation mode.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.
Example	**:AFGenerator:SINGle:FREQuency 2500** **:AFG:SING:FREQ?** Value returned: **"2500.0"**.

:AFGenerator:SINGle:SHAPe

Syntax	:AFGenerator:SINGle:SHAPe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **SINE|RECTangle|TRIangle| POSitivep|NEGativep**. Default is **SINE**.
Description	Sets the waveform of the AF signal to be generated. **SINE** selects a sine wave, while **RECTangle** will generate a rectangular waveform. **TRIangle** selects a triangular waveform. **POSitivep** will generate a sawtooth with a rising (positive) slope, while **NEGativep** switches to a sawtooth with a falling (negative) slope. **Note:** This command will only work in single-tone generation mode. Please refer to the description of the **:AFG:MODE** command for further details regarding the single-tone generation mode.
Query	The query form of this command returns the current setting. A query will return a string, containing the short-form version of one of the predefined expressions explained above.
Example	**:AFGenerator:SINGle:SHAPe TRIangular** **:AFG:SING:SHAP?** Value returned: **"TRI"**.

:AFGenerator:REPLay:DOWNload:FILE

Syntax	:AFGenerator:REPLay:DOWNload:FILE <string>
Parameters	**string** is the name of the audio file to be loaded from the /rapid/wav directory.
Description	Loads an audio file (*.wav) to be replayed. **Note:** Wave files to be replayed over the codec and the RF must be sampled at 8 kHz. Wave files for replay over the audio interface (AF out) must be available with a sampling rate of either 16 or 48 kHz.
Query	The query form of this command returns the current setting. A query will return a string containing the loaded file name.
Example	**:AFG:REPL:DOWN:FILE efr1.wav** Loads file ef1.wav from the /rapid/wav directory to the wave file replay function.

:AFGenerator:REPLay:REPeat

Syntax	:AFGenerator:REPLay:REPeat <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **ON**.
Description	Selects whether the samples shall play once or repetitively.
Query	The query form of this command returns the current setting. A query will return a string containing one of the predefined expressions explained above.
Example	**:AFG:REPL:REP ON** **:AFG:REPL:REP?** Value returned: is **"ON"**.

:AFGenerator:AUXout:STATe

Syntax	:AFGenerator:AUXout:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **OFF**.
Description	Switches the routing of AF signals to the AUX 4 (AUX out) of the Willtek 3100 either on or off. The source of the signal to be output to the auxiliary output connector can be selected using the **:AFG:SPEA:SOUR** command as described below.
Query	The query form of this command returns the current setting. A query will return a string containing one of the predefined expressions explained above.
Example	**:AFGenerator:AUXout:STATe ON** **:AFG:AUXout:STAT?** Value returned: **"ON"**.

:AFGenerator:AUXout:SOURce

Syntax	:AFGenerator:AUXout:SOURce <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **AFIN|AFOUt**. Default is **AFIN**.
Description	This command routes signals coming from a selectable source to the AUX 4 (AUX out) of the Willtek 3100. **AFIN** means that the signal applied to the Willtek 3100 **AF in** connector will be made available at the auxiliary output connector, while **AFOUt** will route the signals from the **AF out** connector to AUX out. This can be signals of the internal AF generator, but also signals coming from the codec of the Willtek 3100. **Note:** Please note that the auxiliary output needs to be switched on first (see command **AFG:SPEA:STAT** above for details).
Query	The query form of this command returns the current setting. A query will return a string, containing the short-form version of one of the predefined expressions explained above.
Example	**:AFGenerator:AUXout:SOURce AFOUt** **:AFG:SPE:SOUR?** Value returned: **"AFOU"**.

:AFGenerator:AUXout:VOLume

Syntax	:AFGenerator:AUXout:VOLume <int1>
Parameters	**int1** is an integer. The minimum value for **int1** is **0**, the maximum value is **100**. The default value for **int1** is **0**.
Description	This command sets the output volume of the external speaker connected to the Willtek 3100 using the AUX 4 (AUX out). **int1** represents a relative volume in percentages of the maximum volume.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	**:AFGenerator:AUXout:VOLume 20** **:AFG:SPE:VOL?** Value returned: **"20"**. This means that the volume at the auxiliary output is at 20% of its maximum.

AFANalyser subsystem

The AFAN subsystem controls the accessible parameters of the AF analyzer. Please note that all commands of this subsystem require the Audio Option to be installed on your 3100.

:AFANalyser:INPut

Syntax	:AFANalyser:INPut <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **AF|AUXin|CODec**. Default is **AF**.
Description	Selects the input signal for the AF analyzer of the Willtek 3100. **AF** means that the signals applied to the test set's AF in connector will be forwarded to the AF analyzer, while **AUXin** means that the signals applied to the test set's **AUX in** input will be forwarded to the AF analyzer. The **AUX in** input is available on the **AUX 4** connector on the back panel of the Willtek 3100 (RF section). **CODec** finally means that the output of the codec (i.e. the AF data transmitted by the mobile under test to the Willtek 3100) will be analyzed. Notes – This command requires the Audio Option to be installed on your Willtek 3100. – RF cannot be signal source and generator destination at the same time. Please check the current setting of the **AFG:OUTP** command before issuing the **AFAN:INP COD** command and vice versa to avoid measurement conflicts.
Query	The query form of this command returns the current setting. A query will return a string, containing the short-form version of one of the predefined expressions explained above.
Example	**:AFANalyser:INPut CODec** **:AFAN:INP?** Value returned: **"COD"**.

:AFANalyser:SAMPlerate

Syntax	:AFANalyser:SAMPlerate <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **KHZ8|KHZ11|KHZ22|KHZ33|KHZ44|KHZ48**. Default is **KHZ48**.
Description	Selects the fixed sampling rate for the AF analyzer's input signals. **KHZ8** means a sampling rate of 8 kHz, **KHZ11** a sampling rate of 11 kHz, **KHZ22** a sampling rate of 22 kHz, **KHZ33** a sampling rate of 33 kHz, **KHZ44** a sampling rate of 44.1 kHz, while **KHZ48** stands for a sampling rate of 48 kHz.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**:AFANalyser:SAMPlerate KHZ22** **:AFAN:SAMP?** Value returned: **"KHZ22"**. This means a fixed sampling rate of 22 kHz.

:AFANalyser:COUPling

Syntax	:AFANalyser:COUPling <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **AC|DC|DCLP**. Default is **AC**.
Description	This command allows to select between AC and DC measurements of the audio signal applied to the AF in connector of the Willtek 3100. If **AC** is selected, the audio analyzer measures the AC signal only, i.e. any DC component in the signal is filtered out before the measurement. With the **DC** setting, the signal is measured directly, including any AC and DC components. If **DCLP** is chosen, the AC components will be filtered out and the Willtek 3100 only measures the DC component of the signal.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**:AFANalyser:COUPLING DC** **:AFAN:COUP?** Value returned: **"DC"**.

:AFANalyser:BALanced

Syntax	:AFANalyser:BALanced <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **OFF**.
Description	Selects whether AF in shall be used as an unbalanced input (setting **OFF**) or as a balanced input (setting **ON**). **Note:** While a balanced signal requires a differential amplifier as an input stage, an unbalanced input doesn't. Balanced signal transmission usually is substantially less sensitive to electromagnetic interference.

Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**:AFANalyser:BALanced OFF** **:AFAN:BAL?** Value returned: **"OFF"**. This means that the AF in socket will be used as a standard input for 'grounded' signals.

:AFANalyser:VRANge

Syntax	:AFANalyser:VRANge <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: AUTO|V30|V3|MV300|MV30. Default is AUTO.
Description	This command sets the range of the expected input voltage on the Willtek 3100 AF in connector. AUTO means that the Willtek 3100 will take sample measurements first and then - based on these measurement results – will decide automatically which setting to use. V30 means that an input voltage of up to 30 volts (rms) is expected and that the resolution of the signal measured will be 100 mV. V3 means that an input voltage of up to 3 volts (rms) is expected and that the resolution of the signal measured will be 10 mV. MV300 expects a maximum input voltage of 300 mV (rms) and delivers a resolution of 1 mV, while MV30 means that the maximum input voltage will be 30 mV (rms) and the resolution will be 100 µV. **Notes** – The better the current input voltage corresponds to the range setting performed with this command, the more precise the measurement results will be. – Whenever fast availability of measurement results is an issue, **AUTO** ranging should be avoided as it is time-consuming.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions as explained above.
Example	:AFANalyser:VRANge V30 :AFAN:VRAN? Value returned: "V30".

:AFANalyser:MRANge

Syntax	**:AFANalyser:MRANge <PredefExp>**
Parameters	**PredefExp** is one of the following predefined expressions: **AUTO|V1|MV100**. Default is **AUTO**.

Description	This command sets the range of the expected input voltage on the Willtek 3100 AUX 4 (AUX in) connector. **AUTO** means that the Willtek 3100 will take sample measurements first and then – based on these measurement results – will decide automatically which setting to use. **V1** means that an input voltage of up to 1 Volt (rms) is expected and that the resolution will be 1 mV. **MV100** expects a maximum input voltage of 100 mV (rms) and delivers a resolution of 100 µV. Notes – The better the current input voltage corresponds to the range setting performed with this command, the more precise the measurement results will be. – Whenever fast availability of measurement results is an issue, **AUTO** ranging should be avoided as it is time-consuming.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions as explained above.
Example	**:AFANalyser:MRANge MV100** **:AFAN:MRAN?** Value returned: **"MV100"**.

:AFANalyser:FILTer

Syntax	:AFANalyser:FILTer <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **NONE|CCITt|CMESsage**. Default is **NONE**.
Description	Selects the type of filter to be inserted between the signal input and the AF analyzer. **NONE** means that there will be no filtering. **CCITt** selects a speech filter as defined by CCITT regulations. **CMESsage** is a speech filter according to US standards.
Query	The query form of this command returns the current setting. A query will return a string, containing the short-form version of one of the predefined expressions explained above.
Example	**:AFANalyser:FILTer NONE** **:AFAN:FILT?** Value returned: **"NONE"**.

:AFANalyser:SPECtrum:RLEVel

Syntax	:AFANalyser:SPECtrum:RLEVel <real1>
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **0.001**, the maximum value is **30.0**. The minimum resolution for **real1** is **0.001**. The default value for **real1** is **1.0**.
Description	This command sets the level of the reference voltage (i.e. the 0 dB line of the AF spectrum display). The physical unit of **real1** is volt if the signal source is **AF in**. Notes – When the signal source is **RF in**, the 0 dB line is interpreted as full scale. – A typical audio standard is 0.775 volts for 0 dB.
Query	The query form of this command will return the current setting. The string delivered back will contain one floating point real number.

Example **`:AFANalyser:SPECtrum:RLEVel 13.0`**
`:AFAN:SPEC:RLEV?` Value returned: **`"13.0"`**.

MS Power Supply subsystem

The PSUPply subsystem controls the accessible parameters of the MS Power Supply Option.

:PSUPply:STATe

Syntax	:PSUPply:STATe <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **ON|OFF**. Default is **OFF**.
Description	Switches the output of the power supply option of the Willtek 3100 either on or off. Please note that this command needs the power supply option to be installed on your Willtek 3100.
Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**:PSUPply:STATe ON** **:PSUP:STAT?** Value returned: **"ON"**.

:PSUPply:LEVel

Syntax	:PSUPply:LEVel <real1>
Parameters	**real1** is a floating point real number. The minimum value for **real1** is **0.0**, the maximum value is **10.0**. The minimum resolution for **real1** is **0.05**. The default value for **real1** is **0.0**.
Description	This command sets the output voltage of the optional power supply. The value specified for **real1** is the output voltage in volt. Please note that this command needs the power supply option to be installed on your Willtek 3100.
Query	The query form of this command returns the current setting. The query will return a string, containing one floating point real number.
Example	**:PSUPply:LEVel 3.65** **:PSUP:LEV?** Value returned: **"3.65"**.

:PSUPply:MEASure:CMAX

Syntax	:PSUPply:MEASure:CMAX <PredefExp>
Parameters	**PredefExp** is one of the following predefined expressions: **A4|A2|A1|MA400**. Default is **A4**.
Description	Selects the range of the maximum current the mobile under test is expected to draw. For a maximum current of up to 400 mA, the **MA400** setting can be used. If the mobile will not draw more than 400 mA, the **MA400** setting is appropriate. In case the maximum power requirement of the mobile is unknown, setting **A4** (the default setting) is recommended as it will allow the mobile to draw a maximum current of 4 A. Please note that this command needs both the power supply option and the current measurement option to be installed on your Willtek 3100.

Query	The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.
Example	**`:PSUPply:MEASure:CMAX MA400`** **`:PSUP:MEAS:CMAX?`** Value returned: **`"MA400"`**.

Publication History

Revision	Comment
0510-100-A	First version.

Willtek Worldwide Offices

West Europe/Middle East/Africa

Willtek Communications GmbH
Gutenbergstr. 2–4
85737 Ismaning
Germany

info@willtek.com

Willtek Communications SARL
69 rue de la Belle Etoile
Bâtiment G
95700 Roissy
France

willtek.fr@willtek.com

Willtek Communications Ltd.
Landmark House, Station Road
Cheadle Hulme
Cheshire SK8 7BS
United Kingdom

willtek.uk@willtek.com

North America/Latin America

Willtek Communications Inc.
25 Eastmanns Road
Parsippany, NJ 07054
USA

willtek.us@willtek.com
willtek.cala@willtek.com

Asia Pacific

Willtek Communications
22, Malacca Street
#09-00, Royal Brothers Building
Raffles Place
Singapore 048980

willtek.ap@willtek.com

Willtek Communications (China) Ltd.
Room 1403, 14/F Cimic Tower,
1090 Century Avenue, Pudon,
Shanghai, 200120
China

willtek.cn@willtek.com

Manual ident no. M 293 111
Manual version 0510-100-A
English

will'tek