4400 Mobile Phone Tester Series
4463 GSM/GPRS System Option, 4468 EDGE System Option

User's guide
Version 6.20

boosting wireless efficiency

Ordering information   This guide is issued as part of the **Willtek 4463 GSM/GPRS System Option and 4468 EDGE System Option**. The ordering number for a published guide is M 293 016. The ordering number for the product depends on the model of the 4400 Series as follows:

Table 1   Order numbers for the 4400 Series

Description	Order number
Willtek 4403 Mobile Phone Tester	M 101 105
Willtek 4405 Mobile Phone Tester	M 101 104
Willtek 4463 GSM/GPRS System Option	M 897 162
Willtek 4462 GPRS Option (for units with GSM already installed)	M 897 159
Willtek 4468 EDGE System Option	M 897 250

Compliance statements   This manual refers to the 4463 GSM/GPRS System Option and 4468 EDGE System Option software. For hardware compliance with any national regulations, please refer to the Getting Started Manual that was delivered with your 4400. The specifications for the product can be found in the data sheet.

4463 GSM/GPRS System Option and 4468 EDGE System Option   Version 6.20

Table of Contents

Chapter 3	**GSM/GPRS Asynchronous Mode Operation**	**129**

Chapter 4	**EDGE Call Mode Operation**	**151**

Chapter 8	**SCPI**	**341**

Publication History   701

About This Guide

Index of the 4463 GSM/GPRS System Option, 4468 EDGE System Option

"Safety Notes" – Read before starting!

"Overview" – Crash course, cabling, menus and levels, fields, parts, help on help

"GSM/GPRS Asynchronous Mode Operation" – Testing a mobile phone in service mode

"GSM/GPRS Call Mode Operation" – Description of the tests and systems which can be tested using call setup

"Tools" – Description of the available tools

"RAPID!" – Introduction, syntax, language elements, error messages

"SCPI" – General introduction and structure

SCPI Command Reference – The commands for the 4463 GSM/GPRS System Option and 4468 EDGE System Option

"Warranty and Repair" – Just in case that your unit should need fixing

> **NOTE**
>
> This manual covers all presently known variants of the Willtek 4400 Mobile Phone Tester Series: 4400M, 4400S, 4403, 4405, 4407. The manual refers to all members of the Willtek 4400 series simply as the 4400.

Purpose and scope

The purpose of this guide is to help you successfully use the 4463 GSM/GPRS System Option and 4468 EDGE System Option features and capabilities. This guide includes task-based instructions that describe how to install, configure, use, and troubleshoot the 4463 GSM/GPRS System Option and 4468 EDGE System Option. Additionally, this guide provides a description of Willtek's warranty, services, and repair information.

Assumptions

This guide is intended for novice, intermediate, and experienced users who want to use the 4463 GSM/GPRS System Option and 4468 EDGE System Option effectively and efficiently. We are assuming that you have basic computer and mouse experience and are familiar with basic telecommunication concepts and terminology.

Related information

Use this guide in conjunction with the following information:

4400 getting started manual, order number M 295 011

Technical assistance

If you need assistance or have questions related to the use of this product or call one of Willtek's technical assistance centers. You can also contact Willtek by e-mail at **customer.support@willtek.com**.

Table 1   Technical assistance centers

Region	Phone number	Fax number, e-mail address
Europe, Middle East, Asia, Africa	+49 (0) 89 996 41 386 +49 (0) 89 996 41 227	+49 (0) 89 996 41 440
Americas	+1 973 386 9696	+1 973 386 9191
China	+86 21 5836 6669	+86 21 5835 5238

Conventions

This guide uses naming conventions and symbols, as described in the following tables.

Table 2   Typographical conventions

Description	Example
User interface actions appear in this **typeface**.	On the Status bar, click **Start**.
Buttons or switches that you press on a unit appear in this **Typeface**.	Press the **On** switch.
Code and output messages appear in this `typeface`.	`All results okay`
Text you must type exactly as shown appears in this **`typeface`**.	Type: **`a:\set.exe`** in the dialog box.
Variables appear in this <typeface>.	Type the new <hostname>.
Book references appear in this **typeface**.	Refer to **Newton's Telecom Dictionary**
A vertical bar | means "or": only one option can appear in a single command.	`platform [a|b|e]`

Table 2   Typographical conventions (Continued)

Description	Example
Square brackets [ ] indicate an optional argument.	`login [platform name]`
Slanted brackets < > group required arguments.	`<password>`

Table 3   Keyboard and menu conventions

Description	Example
A plus sign + indicates simultaneous keystrokes.	Press **Ctrl+s**
A comma indicates consecutive keystrokes.	Press **Alt+f,s**
A slanted bracket indicates choosing a submenu from menu.	On the menu bar, click **Start > Program Files**.

Table 4   Symbol conventions

This symbol represents a general hazard.

This symbol represents a risk of electrical shock.

NOTE
This symbol represents a note indicating related information or tip.

Table 5   Safety definitions

WARNING
Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION
Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

Safety Notes

This chapter provides the safety notes for the 4400 Series Mobile Phone Tester. Topics discussed in this chapter include the following:

- "Safety class" on page xx
- "Before startup" on page xx
- "During test" on page xx
- "During maintenance and repair" on page xx
- "Shutdown when defective" on page xxi

Safety class

The 4400 is designed and tested in line with DIN 57411 part 1 (protective measures for electronic test equipment). The instrument complies with safety class I. It left the works in a perfectly safe condition for operation. To make sure it stays this way and can be operated without any risk, please observe carefully the following notes, which are based on section 17 of DIN 57411 part 1 a.

Before startup

Ensure proper supply voltage

Before powering on, ensure that the operating voltage which is permitted for the 4400 (115 V ±20% or 240 V ±20%) is identical with your line voltage. You do not need to set the voltage range; the 4400 automatically adjusts to the applied (permissible) line voltage.
The power plug of the 4400 may only be inserted in outlets with a ground contact.
Never use extension cables without grounding conductor.

Apply proper grounding

The grounding conductor must under no circumstances ever be interrupted, neither inside nor outside the test set. If there is no grounding through the grounding conductor, the cabinet of the 4400 could become live as the result of a defect. This can make the test set a potential risk.

During test

Apply proper grounding

Always plug in the power socket of the 4400 before connecting a test circuit in order to use the protective effect of the 4400 grounding.

During maintenance and repair

Live parts can be exposed when you open covers or remove components. Connecting parts can also be live.

Live parts with and without supply voltage connected

Before any adjustment, maintenance repair or replacement of parts, the test set must be separated from all voltage sources if it will be necessary to open it. If any jobs have to be performed on the 4400 while voltage is applied, they should only be undertaken by a specialist who is aware of the dangers that are involved. Capacitors in the power supply can still be charged, even though the instrument has been separated from all voltage sources.

> **High voltages after power disconnect**
>
> High and dangerous voltages can still be present in the instrument even when it is being powered from a battery.

> **Use proper fuses**
>
> Defective fuses must be replaced with fuses of identical specifications. Never patch fuses or short the fuse holder.

Shutdown when defective

In the following cases, safe operation is very likely to be no longer possible:

- if the test set exhibits visible damage,
- if the test set will no longer work,
- after a longish period of storage under the wrong conditions (see data sheet),
- following transport under adverse conditions.

If you suspect that it is unsafe to continue operating the 4400, shut it down immediately and pull out all power connectors.
Secure the test set in such way that nobody else can start it up again, in order to protect any third party.
Get in touch with your nearest Willtek service center.

Overview

1

This chapter provides a general description of the 4400 Series Mobile Phone Tester. Topics discussed in this chapter include the following:

About the 4400

The Willtek 4400 Mobile Phone Tester Series has been designed to meet the needs and requirements of service, manufacturing, quality assurance and engineering facilities. For all these areas, accuracy is the key and the 4400 provides exceptional value (superb VSWR and accuracy). For manufacturing, speed is an additional key requirement.

The 4400 features two different operating modes: The **synchronous mode (call mode)** provides call processing with multiband handovers for fast measurements in different frequency bands. The **asynchronous mode (non–call mode)** is dedicated to the board-level alignment, providing generator and analyzer capabilities.

The 4400 series of products provides solutions for:

- CDMA, AMPS (CDMA2000 1xRTT System Option),
- GSM (GSM System Option),
- HSCSD (with Multislot HSCSD Option),
- GPRS (GSM/GPRS System Option),
- EGPRS (EDGE System Option),
- WCDMA/UMTS (WCDMA Non-Call Mode Option, WCDMA Call Mode Option),
- TD-SCDMA (TD-SCDMA Non-Call Mode Option).

The built-in programming tool **RAPID!** allows for easy test automation. Default standard test routines are already stored on the machine, and more enhanced sequences are available (as options).

RAPID! even allows for data input from bar code readers or the control of other external devices. For example, the ability to send AT commands to a cellular phone enables completely automatic testing. Results can be easily documented through screen dumps to an attached printer or file storage. The numerical data of graphs can be read out via GPIB or written to a floppy disk.

Since requirements for the different application areas are different, the 4400 provides a wide range of various options:

The **Audio Option** allows for acoustical measurements and supports dedicated vocoders (speech codecs).

The **Power Supply Option** works as a battery replacement for terminals.

The **Current Measurement Option** allows for easy calculation of standby or talk times.

What's new in version 6.20

WCDMA Option:

- Inner Loop Power Control
- Open Loop Power Control
- Code Domain Specturm with Code Domain Power and Peak Code Domain Error (PCDE)
- Support of all Reference Measurement Channels: 12.2 kbps, 64 kbps, 144 kbps and 384 kbps.

TD-SCMDA System Option (non-call mode):

- Code Domain Spectrum and RX/TX switch

GSM System Option:

Non-residual BER measurement with CRC check.

Features and capabilities

- Based on the multiformat, future-proof platform 4400
- Adaptable to individual needs with options including CDMA2000, GSM/GPRS, EDGE, WCDMA, TD-SCDMA, Audio, Codec, MS Power Supply, MS Current and TCP/IP
- Easy-to-use graphical user interface with external mouse, keyboard and monitor
- Standard SCPI commands for remote control
- Built-in RAPID! for stand-alone automated procedures, allowing the user to easily develop and run programs on the 4400
- DSP platform supports parallel testing of TX, RX and Audio
- Driver software available for Test Stand and Lab Windows CVI

Connecting test leads

Crash course – Starting a test is easy!

Cabling – Connecting a mobile to the 4400

Crash course

To perform a test on the 4400, keep to the following procedure:

1 Switch on the 4400 using the standby button on the front panel.

2 Select the required communication system, e.g. GSM or CDMA.

3 Connect the mobile to the 4400.

4 Switch on the mobile and set up a call.

5 Navigate to the required test menu.

> **Don't exceed the maximum RF power level**
>
> **Before** connecting any RF equipment to the 4400, please check that the maximum input power level is not exceeded. The maximum input power level is indicated on the front panel of the 4400, just besides the RF in/out socket.

Cabling

For precise measurements, we strongly recommend to use an RF adapter cable between the mobile under test and the Willtek 4400. Mobile-specific RF adapter cables are available from the mobile's manufacturer or from Willtek. Please contact your local Willtek representative.

However, should the adapter cable for your mobile not be at hand, the air connection offers a simple-to-use alternative for quick tests. With the optional RF Shield, precise measurements can be achieved without interference.

> **Don't exceed the maximum RF power level**
>
> **Before** connecting any RF equipment to the 4400, please check that the maximum input power level of the 4400 is not exceeded in any case. Otherwise, the highly sensitive input stage may be destroyed immediately. The maximum input power level is indicated on the front panel of the 4400, just besides the RF in/out socket.

Cable connection

The cable connection between a mobile under test and the 4400 requires a mobile-specific RF adapter cable. This RF adapter provides the mobile-specific connector on one end and the Willtek RF–click connector on the other one. Please note that you will need the optional adapter cable 'RF click to N–type' in oder to use the RF adapters on the 4400's N–type RF in/out socket.

Air connection

Using the air connection is a simple method to connect a mobile under test to the 4400 when there is no special RF adapter available. Basically, there are three possibilities to use the air connection.

The simplest possibility is to put an optional antenna on the 4400's RF in/out socket. However, the drawbacks of this solution are the unknown coupling loss and the potential interference.
The coupling loss depends on factors such as the orientation of the mobile towards the 4400, the distance and the frequency.
The possible interference depends on the type of RF radiating equipment being used in the same room or area and on the distance to e.g. a 'real' base station. Consequently, this solution only works for basic functional tests.
Please also note that you may need different antennas for different frequency bands (GSM 900/1800/1900 or cellular/PCS bands).

Using the optional Antenna Coupler, you already have two antennas built into a special casing. Because of the fixed position of the mobile, the coupling loss is much better reproducible and can be fairly compensated using a coupling loss

table. However, there is still no protection against potential interference.
The operating range of the Antenna Coupler covers the complete GSM 900, 1800 and 1900 frequency ranges.
As a standard, the Antenna Coupler comes with an RF-click connector plus an adapter cable 'RF click to TNC'.
Please note that you will need the optional adapter 'TNC to N-type' in order to use the Antenna Coupler on the 4400's N-type RF in/out socket.

If you need to perform reproducible measurements without having the possibility of using a cable, we recommend to work with the optional RF Shield. This is a metal box that covers both the mobile under test and the Antenna Coupler. The RF Shield keeps radio interference away from the air connection. The achievable measurement quality can be as good as when using a cable if the attenuation is compensated with the help of the coupling loss table.
The RF Shield is available with either a TNC-type or N-type connector.

NOTE

The RF Shield Package contains both the RF Shield and the Antenna Coupler.

Using the front and rear panels

Front panel – Get to know the ten sections of the front panel.

Rear panel – Here you will find all about the rear panel of your 4400.

Connectors – The three connectors on the front panel provide the interfaces to the mobile under test. The back panel features all the interfaces you need to integrate the 4400 into your test environment.

The keys on the front panel – Don't be confused about softkeys, marker tabs, function keys, numeric keys, cursor keys, the selection key or mapping the 4400's keys to an external keyboard. Here you will find all you need to know about any key.

Front panel The front panel of the 4400 is divided into ten sections.
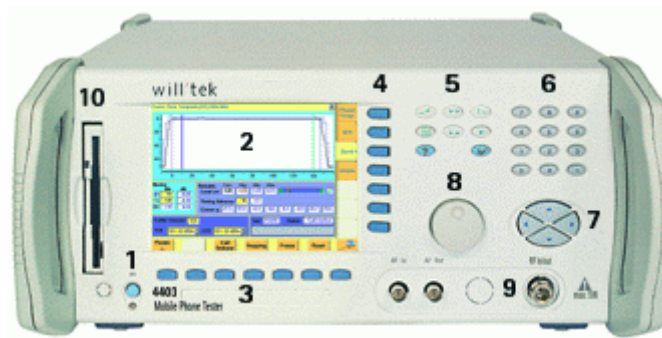
Section 1 – Standby button and MS Power Supply connector

When you turn on the power switch on the back of the 4400, the unit goes to standby mode and the indicator just underneath the standby button is backlit in red.
Push the standby button to switch the 4400 into operating mode. The standby indicator will then turn green.
When you push the standby button again, the unit will go back into standby mode.

> **NOTE**
>
> The power switch is located on the rear panel of the 4400. A dark standby button indicates that the power switch is off.

The MS Power Supply connector is used in conjunction with the MS Power Supply and Current Measurement Options. It provides supply voltage and current for the mobile under test.
The cable for this connector is delivered with the MS Power Supply Option.

Section 2 – Display

The display is the graphical user interface of the 4400. It will show you menus with entry fields and display fields. The display also labels the software-driven softkeys (see section 3) and marker tabs (see section 4).

Section 3 – Softkeys

Softkeys call specific functions that vary with the menu displayed. For instance, when you switch on the 4400, the softkeys will allow you to select the kind of system you want to work with (e.g. GSM or CDMA).

Section 4 – Marker tabs

Like the softkeys, the marker tabs are software-labelled keys used for selecting a test (e.g. Phase/Frequency test), for navigating through this online help, or for loading different tools (e.g. Analyzer).

Section 5 – Function keys

Function keys are used for general operation. They have a fixed function. The **ENTER** key, for instance, confirms changes on entry fields.

Section 6 – Numeric keys

The numeric keys are mainly used to enter values on entry fields.

Section 7 – Cursor keys

The **LEFT** and **RIGHT** cursor keys are used to select the required entry field of a menu. A selected entry field is inverted and therefore easily visible on-screen. With the help of the **UP** and **DOWN** cursor keys, values on entry fields can be changed.

Section 8 – Selection key

The selection key is actually more a wheel. It is used to quickly change values on entry fields.

Section 9 – Connectors

The connectors on the front panel provide the interfaces for the mobile under test.
The connection of a mobile is outlined in section Cabling.
The detailed description of all connectors on both the front and back panel of the 4400 can be found in section Connectors.

⚠️ **HAZARD**

Before connecting any equipment, please make sure that the maximum input levels of the connectors are not exceeded. Otherwise, the highly sensitive stages of the 4400 might be destroyed immediately.

Section 10 – Floppy drive

As with all floppy drives, the floppy disk will only fit in one direction. Please insert the floppy disk carefully. Do not use any force as this could damage both, the floppy disk and the floppy drive.

Rear panel

The rear panel of the 4400 is subdivided into three main sections.
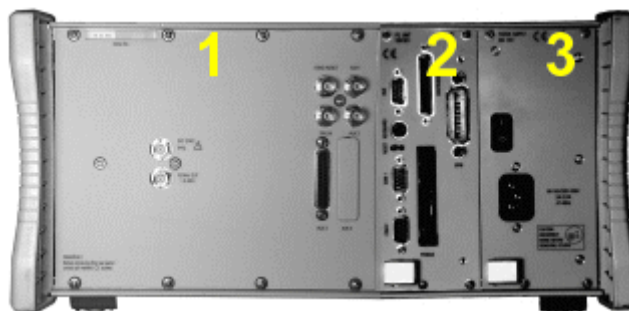
Section 1 – The RF/IF/AF section

The RF/IF/AF section provides several interfaces that may support your tests. Most of these interfaces are related to synchronization and triggering.
Section Connectors gives a detailed overview of all the connectors, their functionality and technical specifications.

Section 2 – The PC unit

The 4400 is powered by an industrial PC. Therefore, you will find all the standard PC interfaces on the back of the 4400.
If you need to know anything regarding the PC interfaces, please refer to Connectors.

Section 3 – The power supply

In this section, the power switch and the power connector are located.

NOTE

The power switch has to be **on** before you will be able to start the 4400 using the **STANDBY** button.

⚠ **HAZARD**

Before connecting the 4400 to the power line, please check that the power available on your supply line is within the operating range of the 4400's power supply as indicated on the back of the unit and in the getting started manual.

Connectors

Connectors on the front panel   There are three connectors on the front panel:

AF in — BNC socket. Input for AF (audio) signal. This input is only used when the Audio option is installed.
This input can be used either as a balanced or as an unbalanced input (software switch).
The input impedance is approx. 250 kΩ/20 pF.
The usable AC frequency range stretches from 30 Hz to 20 kHz.

⚠ **HAZARD**

The **absolute maximum input voltage is ±40 V DC or 30 V (rms) AC**. An AF input voltage of more than ±40 V DC or 30 V (rms) AC may result in an **immediate destruction** of the highly sensitive AF input stage of the 4400. Willtek will not accept liability for any damage of the input stage due to overload.

AF out — BNC socket. Unbalanced AF (audio) signal output of the 4400. Used only when the Audio option is installed.

Best performace is achieved on a 600 Ω load.

The maximum output voltages are 4 V (rms) for sinusoidal signals or 11 V peak-to-peak for all other signals.

RF in/out — RF signal input for transmitter measurements and output for receiver measurements. It depends on the current system test whether the 4400 measures an input signal, generates an output signal or both.

> ⚠ **HAZARD**
>
> The **absolute maximum input power is 5 W (37 dBm)**. An RF input power of more than 5 W may result in an **immediate destruction** of the highly sensitive RF input stage of the 4400. Willtek will not accept liability for any damage of the input stage due to overload.
> **Note:** There is no time limitation for applying the maximum input power.

Connectors on the rear panel

The rear panel is divided into three main sections. These are from left to right: the RF section (1), the PC unit (2) and the Power supply (3). Each section provides one or more connectors for specific functions. The connectors are described from left to right and from top to bottom.
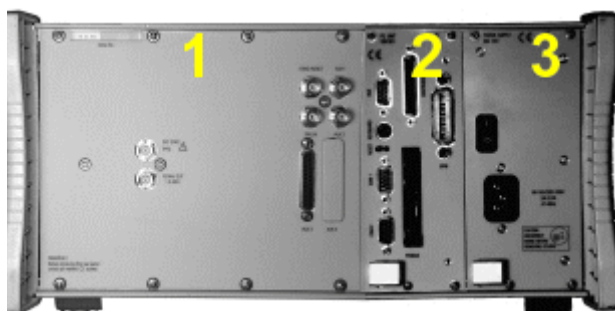
Connectors on the RF section
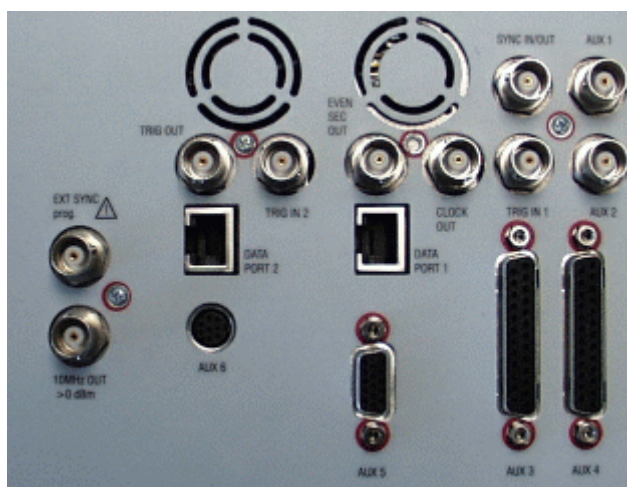
Not all of the connectors may be fitted in your 4400. The connectors actually available depend on your system configuration, in particular the installed system options.

EXT SYNC — BNC socket. Input for external reference signal. If a high-precision reference signal of 5 MHz, 10 MHz or 13 MHz is connected here, the internal 10 MHz reference signal of the 4400 will not be used.
Input level: approx. 0 dBm
Frequency tuning range: approx. 1 ppm (10E-6) (equals 10 Hz at 10 MHz)
Input resistance: approx. 50 Ω

10 MHz OUT — BNC socket. Output of the 4400's internal 10 MHz reference signal. With the help of this output, the internal time base of the 4400 can be used to synchronize external equipment.
This output is very useful if the optional oven-controlled oscillator (OCXO) is installed. More information on the OCXO can be found in section Accessories and options.
Output power: approx. +5 dBm
Output resistance: approx. 50 Ω

SYNC IN/OUT — BNC socket, providing a CMOS-TTL signal.
The signal on this connector allows the GSM-specific synchronization with external units. The connector is currently configured as **Sync out**. The 4400 outputs its internally generated frame synchronization signal. This signal is TTL-low for (downlink) time slots 0 to 2 and TTL-high for time slots 3 to 7.

NOTE

In GSM, the timing of up- and downlink is shifted by three time slots to avoid that a mobile has to receive and transmit during the same time slot. The time difference between the falling and the rising edge of the Sync out signal reflects exactly that time shift.

TRIG IN — BNC socket, expecting a CMOS-TTL signal.
A trigger signal applied on this connector triggers all GSM-related RF measurements. The 4400 allows you to select between triggering on the rising or falling edge of the signal connected here.
The main application for this connector will be measurements in non-call mode at very low power levels, e.g. asynchronous measurements.

TRIG IN 2 — BNC connector for future use.

TRIG OUT — BNC connector for future use.

NOTE

For GSM the 4400 can internally trigger its asynchronous measurements as long as the RF power level transmitted by the mobile is above −30 dBm. Only in case you are working with lower RF power levels, it is necessary to supply an external trigger signal on this input.

AUX 1 — Reserved for later use. Please do not connect anything here.

AUX 2 — Reserved for later use. Please do not connect anything here.

AUX 3 — Reserved for later use. Please do not connect anything here.

AUX 4 (AUX out) — Using the Audio Option of the 4400, audio signals can be made available on this output, for example to connect the 4400 to an external loudspeaker. The output is available on **pins 8 and 9** of this connector.
The output impedance is 8 Ω; the maximum output power level is 1 W.

> ⚠ **HAZARD**
>
> AUX out is a balanced output. This means that the voltages provided are **floating**.
> Never short-circuit these two pins and never connect them to ground as this could result in an **immediate destruction** of the AUX out output stage of the 4400.

AUX 4 (AUX in) — This input can be used to pick up audio signals with a microphone and to use those signals for audio measurements.
The auxiliary input is available on **pins 12 and 13** of this connector.
This unbalanced input is only used when the Audio option is installed.
The input impedance is < 1 kΩ.
The usable AC frequency range stretches from 30 Hz to 20 kHz.

> ⚠ **HAZARD**
>
> The **absolute maximum input voltage is 1 V (rms) AC.** An AF input voltage of more than 1 V (rms) AC may result in an **immediate destruction** of the highly sensitive AUX in input stage of the 4400. Willtek will not accept liability for any damage of the input stage due to overload.

AUX 5 — This is a diagnostic port without any user application.

AUX 6 — For future use.

EVEN SEC OUT — This is a CDMA-related periodic signal. The signal on this BNC connector is TTL, with a period of 2 seconds. A negative pulse occurs every two seconds, with a pulse width of approximately 51 nanoseconds.

CLOCK OUT — This BNC connector transmits a periodic TTL CDMA clock signal. Please refer to sections CDMA Call Mode and CDMA Non-call Mode in the CDMA user's guide for more information.

DATA PORT 1 — This RJ45 connector is a test port for CDMA without any user application.

DATA PORT 2 — RJ45 connector for future use.

Connectors on the PC unit

VGA — Standard connector for a VGA monitor.

> **NOTE**
>
> The resolution of the 4400's video output is set to 640 x 480 pixels and can-not be altered.

Keyboard — Standard PS/2 connector for a PC keyboard.

> **NOTE**
>
> An external keyboard is highly recommended for RAPID! programming.

COM 1 — Standard 9-pin D-sub connector for serial interface 1. Usually used to connect a standard PC mouse.

> **NOTE**
>
> A mouse is highly recommended for RAPID! programming.

COM 2 — Standard 9-pin D-sub connector for serial interface 2.

CENTRONICS — Standard parallel port to connect a PC printer (LPT 1).

PCMCIA — These two PCMCIA slots are reserved for LAN interfacing and for future applications.
The slot takes either two Type 1 cards or one Type 2 card.

GPIB — **G**eneral **P**urpose **I**nterface **B**us according to specification IEEE-488.2.

Connector on the power supply
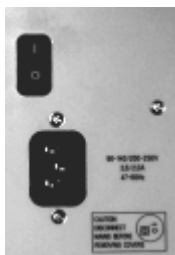
Power in — Standard IEC 320 power connector (3 pins). Please make sure that the power cable is properly grounded.

> **HAZARD**
>
> **Before** connecting the 4400 to the power line, please check that the power available on your supply line is within the operating range of the 4400's power supply as indicated on the back of the unit and in the getting started manual.

The keys on the front panel
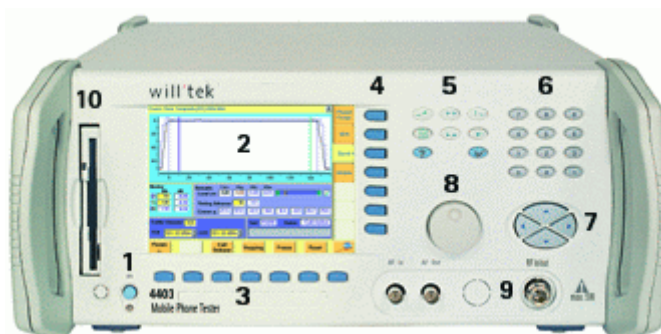
There are six different groups of keys. In the photo of the front panel above, they are marked with numbers 3 through 8.

– The softkeys (3)
– The marker tabs (4)
– The function keys (5)
– The numeric keys (6)
– The cursor keys (7)
– The selection key (8)

Notation of keys in this manual and online help

Hard keys, such as the **ENTER** key, appear in capital letters.

Either a graphical image of the softkey is used, or its label is in bold print, e.g. **Param...**.

External keyboard and mouse

There is the possibility to connect a standard PC keyboard and/or a standard serial PC mouse to the 4400. When an external keyboard is connected, the keys on the front panel of the 4400 and the keys on the external keyboard may be used in parallel.

In order to give you access to the 4400-specific keys like the **TOOLS** function key, those keys are mapped to the external keyboard. The mapping table can be found in section Keyboard mapping table.

The connectors for both the mouse (COM 1, Sub-D 9) and the external keyboard (Keyboard, PS/2) are located on the back panel of the 4400 in the PC section. Section Connectors provides you with all the information about those connectors and their locations.

The softkeys　Softkeys call specific functions that vary with the menu displayed. The specific function is written in the last two lines of the display, just above the physical key. This is what we call the label of the softkey.

Example: In the Welcome menu of the 4400, the softkeys will allow you to select between measurements in call mode and those in non-call mode.

Notes on softkeys

–　When the label of the softkey ends with an ellipsis (...), this softkey will take you to a new menu (of a lower level), where you can enter e.g. additional parameters.
　　To return from a lower menu level, push the **ESCAPE** function key.
　　Example: Softkey **Param...** will take you to a new menu, where system parameters can be set.

–　When there is a test in progress, some softkeys will be **blocked**. They turn to gray to indicate that the previously started test has to be either completed or aborted, before a new selection with the softkeys is possible.
　　Example:

While softkey **STOP** could be pushed any time, softkey **Paging** is not accessible as long as the current test or procedure is in progress.

–　While the Online help is on display, the softkeys change to navigating aids and help you to find your way through the Online manual.

The marker tabs

When there are several menus of the same level, e.g. different system tests, the seven marker tabs make it easy to switch between them.
When the Online help is on display, the marker tabs provide the links to related pages.

Selected, deselected and blocked marker tabs

Marker tabs can be selected, deselected or blocked.
A **selected marker tab** is highlighted in yellow and indicates that the current menu has been selected.
Deselected marker tabs are orange to indicate that they could be selected any time.
When there is a test in progress, some or all marker tabs will be **blocked**. They turn to gray to indicate that the previously started test has to be either completed or aborted, before a new selection with the marker tabs is possible.

Marker tab [MS Info], not selected

Marker tab [MS Info], selected

Marker tab [MS Info], blocked

Notation of marker tabs in this documentation

Wherever this Online help manual refers to a marker tab, either the marker tab is shown graphically, or it is indicated by setting its name between square brackets, e.g. [MS Info].

The function keys

With the help of the function keys, you have instant access to important or useful functions of the 4400. Some of the function keys help you to navigate through the current menu.
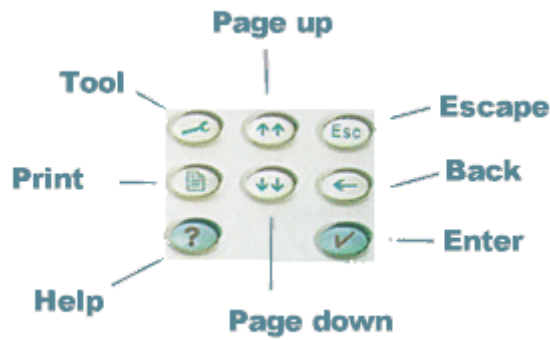
Meaning of the function keys

Symbol	Name	Description
	TOOL	Calls the tools menu. From there, you start the special tools like the Audio measurements or the MS Power Supply.
	PGUP, PGDN	A push selects the first entry field of the previous or next area. When the Online help is on display, these keys call the previous or next page of the Online help.
	ESCAPE	A push on this key takes you one menu level up. Should this cause a change of a parameter, you will be prompted to either confirm or discard your change in menu levels.
	BACK	Deletes the character left to the cursor on an entry field.
	ENTER	Confirms a value entered on an entry field.
	HELP	Calls the Online help on-screen. Once that you are done with the help screens, press [Escape] to return to the previous screen. For more information about the online help system, see section Help on Help.
	PRINT	Prints the current screen to the attached printer or generates a screenshot file on the floppy disk. For more detailed information, see section Parallel Port Settings.

The numeric keys

With the help of the Numeric keys, you enter parameter values on entry fields.

Important function keys mainly used in combination with the numeric keys

deletes the character left to the cursor on an entry field.

confirms all the changes or entries.

The cursor keys

With the help of the (**LEFT**) or (**RIGHT**) cursor keys, you navigate the cursor from one entry field to the previous or next one.

The (**UP**) or (**DOWN**) cursor keys allow you to increase or decrease the parameter value on a numeric entry field or to step through the predefined settings of a selection field.

The selection key

This wheel is used to quickly change values on entry fields.

When the cursor is positioned on an Entry field (the Entry field is then inverted), a turn on the selection key directly changes the value. Any change made with the selection key is instantly confirmed. It is not necessary to push the **ENTER** key

Altering numeric values: navigate to the required entry field.
Clockwise rotation (i.e. to the right) of the selection key increases the value, counterclockwise rotation (i.e. to the left) decreases it.

Changing the setting of selection fields: navigate to the selection field.
The available predefined settings are highlighted one by one when you slowly
turn the selection key clockwise or counterclockwise.

Keyboard mapping table   Every key of the 4400 is mapped to a specific key or key combination of an
external PC keyboard.

Softkeys

From left to right, the seven softkeys are mapped to the key combinations
SHIFT+F1 to **SHIFT+F7**.

Marker tabs

From up to down, the seven marker tabs are mapped to the key combinations
CTRL+F1 to **CTRL+F7**.

Function keys

4400 front panel	Keyboard
HELP	**F1**
TOOL	**F2**
PRINT	**F3**
PGUP	**PAGE UP**
PGDN	**PAGE DOWN**
ESCAPE	**ESC**
BACK	**BACKSPACE**
ENTER	**ENTER**

Numeric keys

Mapped to the numeric keys on the standard PC keyboard.

> **NOTE**
>
> Before using the numeric keypad of the external keyboard, please check that
> the **NUM LOCK** is switched on.
> The following specific keys are mapped to:
> **±: -** i.e. the minus key
> **.: .** i.e. the dot key

Cursor keys

4400 front panel	Keyboard
LEFT	SHIFT+TAB
RIGHT	TAB
UP	ARROW UP
DOWN	ARROW DOWN

Selection key

F11 for counterclockwise turn, i.e. to the left,
F12 for clockwise turn, i.e. to the right

Menus and levels

Menus

The entire area of the display is what we call a **menu**. Any time you switch on your 4400, it first displays the start or Welcome menu. The Welcome menu is the highest menu level of the hierarchically structured user interface.

With the help of the marker tabs (on the right-hand side of the display), you select between the different system menus (e.g. [GSM] or [CDMA]).

The softkeys (just below the display) give you instant access to the various test menus (e.g. **Calls & Meas.** or **Gen/Ana**).

Levels

The user interface of the 4400 is organized in a hierarchical order. This means that there are several levels.
The Welcome menu is at the top level. As you go lower in menu levels, you will reach more specialized or detailed functions.

– When there are several menus of the same level, you can select between them using the marker tabs on the right-hand side of the display.

– To navigate to a menu of the next lower level, push any one of the softkeys.

– When you would like to return to the next level above, push the **ESCAPE** function key.

Example: You would like to go to the GSM Burst Zoom menu.
You can use either the Calls & Meas. or the Gen/Ana menus. For tests in call mode, i.e. involving a call setup, simply push the **Calls & Meas.** softkey from the Welcome menu. This will take you to the test menu where you can select between the various tests (this is the so-called test level).
Now you push the third marker tab (from the top) and the 4400 will display the Burst test menu.
The **Zoom** softkey of the Burst test menu leads you to the Zoom menu.

The areas of a test menu

Each test menu is divided into different color-coded areas, each providing different functions. These functions are described in detail in the test sections of the related system.

The figure below gives you a first overview. The example we have chosen is the Phase/Frequency test menu of the GSM system, but the principle is the same for all other test menus.
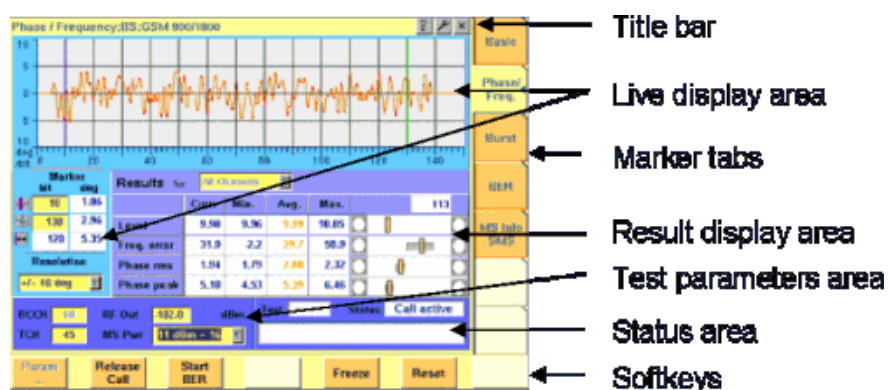
– The **title bar** shows the name of the menu you are currently working in. In the example above, this is the menu testing the phase/frequency character-istics of a mobile in a GSM 900/1800 system.

– The **live display area** shows the measurement results in real time, whether numerical or graphical. To make the live display area easy to identify on the display, it is placed on a light blue background.

– The area with the purple background is the **result display area**. It shows the statistical analysis of the measurement results and the bar display.

– The two lower areas of the menu are both on dark blue background. The left one is the **test parameters area**. On this area, you will find entry fields that allow you to instantly alter relevant test parameters without being forced to navigate to some parameters menu first. The area on the right-hand side is called the **status area**. Here, the 4400 tells you if the current test has passed or failed. Furthermore, the signaling status will be shown in this area.

– The **softkey area** forms the bottom of a menu. With the help of the seven softkeys you can directly call a system parameters menu, start special test functions or zoom the test display.

– The **marker tabs area** on the right-hand side of the menu carries the labels of the seven marker tabs. They provide access e.g. to different tests.

Menu fields

In the menus of the 4400, we use two different types of fields:

– An **Entry field** is what we call a field that is accessible by the cursor and accepts an entry. Entry fields are mainly used to enter test parameters. There are entry fields that only accept numeric inputs (like e.g. a channel number) and there are entry fields that accept alphanumeric data (like e.g. an SMS text).
There are also two special forms of entry fields: the so-called **selection fields** where you may select one of several predefined possibilities (e.g. what bit class(es) you want to use for a bit error rate measurement) and the check boxes.
Check boxes are like an on/off switch. You can't enter values, you can just select or deselect them.
All entry fields have a yellow background to make them easy to allocate on-screen. The only exception to this rule are the check boxes.

– **Display fields** show the results of the performed measurements. Again, there are different types of display fields as measurement results can be purely numerical (e.g. RF power) or they can be alphanumerical (e.g. the serial number of a mobile). The **bar display** also is a display field.
To make display fields easy to distinguish from entry fields, they always have a white background.

Entry fields The Willtek 4400 offers you different types of entry fields:

– **Alphanumeric**, **hexadecimal** or **numeric** entry fields accept text or numbers respectively.

 Example: **TCH** – Here you enter the number of the traffic channel the measurement shall be performed on.

– **Selection fields** are entry fields that offer you a choice from a list of predefined values. Click with the mouse on the triangle to see the available options, or press the **UP** or **DOWN** cursor key.

Example: **TCH Type** – This selection field enables you to choose a specific TCH type from a list.

– **Check boxes** only need a simple yes or no. The 'yes' is indicated by a green check box. To activate or deactivate the function, click on it with the mouse or go the field and press **ENTER**.

Example: The **Call on Loc. update** check box is unchecked. This means that the function will not be performed.

– **Radio buttons** allow you to select between different options, similar to selection fields.

Example: Of the audio generator radio buttons in the picture, the sine function has been selected and the other functions are inactive. By selecting the rectangle function, sine would automatically be deselected.
Select a function by clicking on the appropriate radio button, or move to the radio button with the cursor keys and press **ENTER**.

Navigating to an entry field

Jump to the required area using the function keys **PGUP** or **PGDN**.

Select the entry field you want to make changes on with the help of the **LEFT** or **RIGHT** cursor keys.

The selected entry field is inverted to make it easily visible on-screen.

Entering values on numeric entry fields

There are two ways how to work with numeric entry fields: either you enter the value using the 4400's numeric keys or you use the selection key. The achievable results are basically the same. However, there is no entry discard function when you use the selection key.

– Enter the required value using the numeric keys of the 4400.
While the entry field is open, entered values can also be edited using the **BACK** function key.
To **confirm** the changes, push the **ENTER** function key. If your entry should be out of the possible data range, the 4400 will prompt you with `Data out of range` in the status area of the current menu.
To **discard** the changes, just leave the entry field using the **LEFT** or **RIGHT** cursor keys.

– Instead of typing in a new value, you may also turn the selection key to increase or decrease the current value of the entry field. The **UP** or **DOWN** cursor keys also increase or decrease the value.
Note: The cursor keys have a built-in 'autorepeat function'. This means that the value on the entry field will be incremented respectively decremented by one step on a short push or by a couple of steps if you push the key for a longer period of time.

> **NOTE**
>
> When you use the selection key or the cursor keys to change a parameter value on an entry field, the changes are confirmed automatically. It is not required to push the **ENTER** key.

Entering values on hexadecimal entry fields

Where an entry in hexadecimal notation (0 to 9, A to F) is expected, the marker tabs change to the letters A to F. You can enter data with the numeric keys (0 to 9) and with the softkeys (A to F).

Entering values on alphanumeric entry fields

On alphanumeric entry fields as e.g. **SMS to mobile phone**, texts are best entered using an external keyboard. Of course, it is also possible to enter alphanumeric values directly on the 4400:
As soon as the entry field is opened, the marker tabs change to groups of letters and symbols. The lowest marker tab incorporates the "shift" function and allows you to toggle between lowercase and uppercase letters.

As soon as you select a group of letters or symbols by pushing the corresponding Marker tab, the Marker tabs will change again to single letters or symbols. Push the Marker tab of the desired letter or symbol. That one will appear on the entry field and the Marker tabs will change back to groups of letters and symbols. While the entry field is open, entered values can also be edited using the **RIGHT** function key.

Entering values on selection fields

Again, there are two ways how to work with selection fields: either you change the selection using the 4400's cursor keys or you simply turn the selection key. The achievable results are basically the same. However, there is no entry discard function when you use the selection key.

– Select the required value using the **UP** or **DOWN** cursor keys.
  To confirm the changes, push the **ENTER** function key.
  To discard the changes, just leave the entry field using the **LEFT** or **RIGHT** cursor keys.

– Instead of selecting a new value with the cursor keys, you may also turn the selection key.

NOTE

When you use the selection key to change a parameter value on a selection field, the changes are confirmed automatically. It is not required to push the **ENTER** key.

Switching check boxes

Any push on the **ENTER** function key toggles the currently selected check box. This means that an unchecked check box will be switched to 'on' or 'yes' with the first push and back to unchecked with the second push.
Unchecked boxes are indicated in gray; checked ones are highlighted in green.

Display fields The 4400 offers you two different types of display fields:

(Alpha)numeric display fields

These kind of display fields are used to show test results as numeric values, as strings (text) or as graphical display of numeric values (e.g. the shape of a received burst within a template).

The bar display
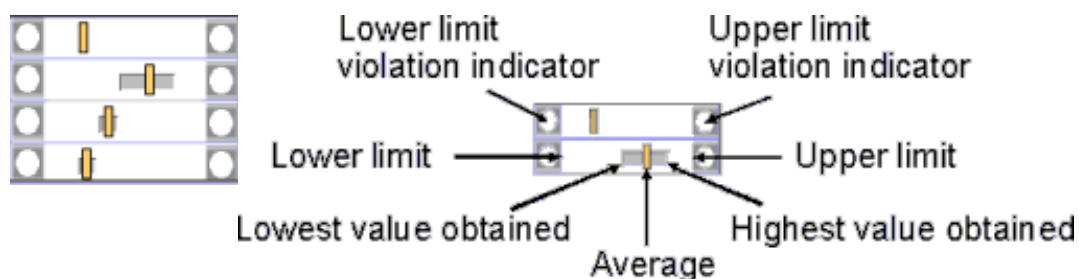
The bar display is used to display measurement results that are obtained by running a specific test a certain number of times. It shows a couple of important values on a relatively small area:

– The large **white area** indicates the range between the lower and the upper limit of the specific test.

– The **dark gray** section within the white area shows the range of all measurement values obtained from the minimum to the maximum.

– The small **orange rectangle** marks the average value.

– The two **limit violation indicators** are white as long as no violations of the test limits occur.

– As soon as a measurement value obtained is smaller than the lower limit, the lower limit violation indicator will turn to red.

– The same will happen with the upper limit violation indicator when a measurement value exceeds the acceptable range.

NOTE

To reinitialize the statistics, push the **Reset** softkey.

Help on Help

This section describes how to work with the Online help. Here you will find all about

– Navigating help,

– Using the online help on an external PC.

Notation rules

While working with the Online help, you will have to push softkeys, marker tabs, or other keys. To keep things as clear as possible, the online manual either uses the graphical image of the item as you see it on-screen or it uses the following notation rules:

Calls & Meas.	The label of a softkey is indicated like this.
[RF Gen]	The label of a marker tab is indicated in square brackets.

ESCAPE	The name of function keys, numeric keys or cursor keys is indicated in this type of letters.
Channel	Entry fields are indicated in normal letters.

Navigating help
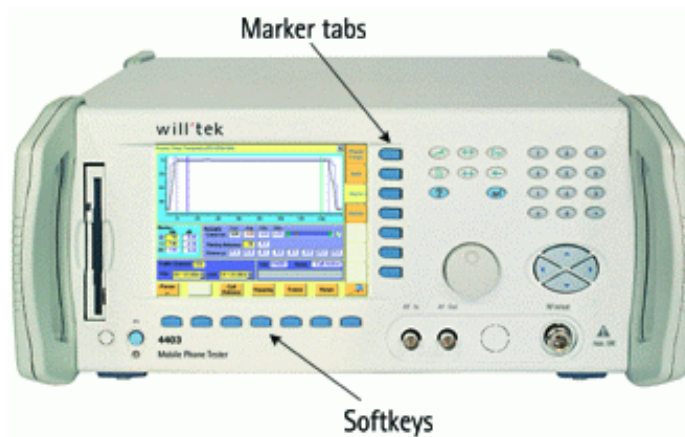
With the help of both the marker tabs and the softkeys, it is easy to navigate through this Online help.

Marker tabs – the links

This Online manual is a so-called hypertext document. It contains **links**. Those links allow you to jump to a different page where more detailed information about a specific topic is provided. On a PC, you would simply click the link. On the 4400, you can use the mouse or the marker tabs for 'clicking' the links.

Example: This page provides links to the documents explaining the softkeys and marker tabs in detail. Use the marker tabs to directly jump to the related pages.

Softkeys – the quick navigators

Using the softkeys you quickly navigate through a chapter or call its index. There are five softkeys available:

Top	This softkey calls the first page (i.e. the page directly following the index page) of the chapter. If that first page is already on display, nothing will happen.
<<	A push on this softkey takes you to the previous page of a chapter. When you push this softkey on any index page, you will get the last page of the previous chapter on-screen.
>>	Calls the subsequent page of a chapter. From the last page of a chapter, this softkey will take you to the index page of the subsequent chapter.
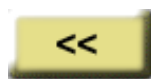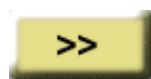
Index	A push on this softkey takes you to the index page of the current chapter. This index page shows the contents of the chapter along with links to the different topics. When you push **Index** on an index page, then the index page of the next level up will be shown.
Back	Takes you back to the help page shown last before the current page.

Using the online help on an external PC

Installing the online help on a PC

Standard PCs offer a much larger screen diameter compared to the 4400. Therefore, it might be more convenient for you to have the Online help running on a PC. Furthermore, your PC would usually be connected to a printer, so printing pages or chapters of the Online manual is easier when working on a PC.
In order to install the Online help on a PC, you need to have the Online manual on disk. This can be obtained directly from Willtek.

NOTE

The Online manual contains a number of subdirectories. Please make sure that those subdirectories are copied to your PC as well. Otherwise, you may experience difficulties getting access to specific images or descriptions.

To start the Online help on your PC, load the file called `frames.htm`. You will find this file in the top level directory of the Online help.
You will be provided with the same navigating aids as on the 4400: the lower frame will show five buttons, having the same functionality as the navigating softkeys on the 4400.

NOTE

If you should explore any problems to navigate with the help of the buttons in the lower frame, load the file **index_e.htm** instead. It is located in the same directory. Loading this file, there will be no specific navigating aids – but you still can use the previous/back functionality of your browser.

Navigating through the help pages

Links instead of marker tabs

On the PC, you may simply click the links on a page like on any Internet web page. For easy identification, these links appear in blue and are underlined. The mouse pointer will change as soon as you position it on any link.

Buttons instead of softkeys

Working with Internet Explorer or Netscape Navigator, you have the same functionality at your disposal as on your 4400.

On the PC screen, you will see two frames. The upper and larger one shows the information, the lower one provides the buttons like on the 4400.
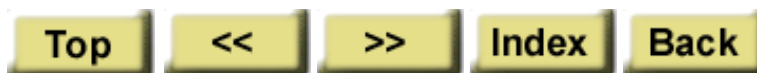
Each section of the manual can be printed with the printing function of the web browser.

Accessories and options

Your 4400 is a highly integrated test set that allows you to perform most of the tests and measurements required in a standard production environment already with its basic configuration. However, there are much more areas of application for your 4400. It can even be used in R&D, for high-level service or for training your staff on digital radio systems.

Willtek has developed a broad range of accessories that help you perform specific tests or measurements, to easily connect mobiles to the 4400 and much more. Please find below a short overview of the available accessories and options, and the applications they have been designed for. Should you miss something in the list below, please check with your local Willtek representative. Due to ongoing product development and improvement, the turnkey solution to your requirement may already be available.

All the latest developments for your 4400 can also be found on the web at www.willtek.com.

> **NOTE**
>
> Specifications are used in this manual for illustration purposes only. Therefore, the specifications of the product delivered may be different from the specifications shown here. However, all relevant specifications for the 4400 can be found in the test set's data sheet. Should that data sheet not be at your disposal, please ask your local Willtek contact for a copy or feel free to download it from Willtek's web site.

Expanding the 4400's measurement capabilities

Audio measurements

In its basic configuration, the 4400 does not support audio measurements. In order to provide audio functions, you need to install the 4470 Audio Option. It provides your 4400 with both an audio generator and audio analyzer.

Additional optional features for audio testing are available for the GSM and GSM/GPRS System Options: The 4471 Basic Codec Option and the 4472 Codec Extension Option allow you to convert the digital data exchanged between the mobile and the 4400 into audible sound.

As you know, GSM currently uses three different codecs.

Full Rate is what GSM started with and what is still the most common codec type. To convert Full Rate into audible sound, the Codec Basic Option is all you need.

Half Rate is supported by many mobiles, although the network operators are hesitant to use this feature because they are afraid of loosing subscribers due to poor speech quality.

Enhanced Full Rate is a newer codec algorithm that provides better speech quality than Full Rate. To convert Half Rate or Enhanced Full Rate into audible sound, you will need the Codec Extension Option in addition to the Basic Codec Option.

Option	Willtek order number
4470 Audio Option	M 248 360
4470 Audio Option for 4400 with CDMA2000 only	M 248 653
4471 Basic Codec Option	M 248 364
4472 Codec Extension Option	M 897 156

Audio measurements on the 4400

High-precision time base

In its basic configuration, the 4400 is equipped with a TCXO 10 MHz time base. The specified accuracy is approx. 1E-6 or 1 ppm (10 Hz). The OCXO option provides an increased accuracy of approx. 5E-8 or 0.05 ppm (0.5 Hz).

Option	Willtek order number
4471 OCXO option	M 214 028

Multislot measurement capabilities (HSCSD) (for GSM, GSM/GPRS and EDGE only)

HSCSD (= High Speed Circuit Switched Data) enhances standard GSM with the bundled use of several time slots within a GSM frame for data transmission. Theoretically, the maximum data transmission rate in HSCSD is n * 14.400 bps, where n is the number of the bundled time slots.
In case of the 4400, the maximum for n equals 4; this means that the 4400 can transmit and/or receive on four time slots per frame.
The tool to perform tests on HSCSD mobiles is called Multislot/HSCSD Option.

Option	Willtek order number
4461 Multislot/HSCSD Option	M 897 158

ACPM or Output RF Spectrum Option (for GSM and GSM/GPRS only)

The Adjacent Channel Power Measurement (ACPM) Option provides the ETSI-specified measurements on Output RF Spectrum, both due to modulation and due to switching transients. Please refer to the 4400 data sheet for specifications. The measurements are explained in the sections about ACPM Modulation Spectrum Measurements and ACPM Transient Spectrum Measurement, respectively.

Option	Willtek order number
4475 APCM Option	M 897 163

NOTE

This option is already included with the 4400M.

NOTE

Adjacent channel power measurements (ACPM) are already included with the CDMA2000 1xRTT System Option.

AM Signal Generator (for GSM and GSM/GPRS only)

The Amplitude Modulation (AM) Signal Generator is required by some manufacturers for testing and alignment of mobile phones. The functionality is explained in the section about the non-call mode.

Option	Willtek order number
4481 AM Signal Generator Option	M 897 165

Little aids for speeding up tests

Operating the 4400 over a LAN with the TCP/IP Option

The 4478 TCP/IP Option allows you to connect the 4400 to a LAN. This method has multiple benefits:

– It allows you to control the 4400 from an external PC (resolving the need for a GPIB card in the PC).
– You can store result files on a remote PC from within your RAPID! application or from the file manager (see sections "File menu" on page 265 and "Commands for input/output handling" on page 306).
– You can access files on the 4400 from a remote PC (requires additional NFS client software on the PC).
– You can speed up software updates and avoid bothering with floppy disks. See section "Initiating a software update from a remote PC" on page 150.

The 4478 TCP/IP Option adds remote access capabilities to your 4400 from the software side. The option includes an ethernet card to physically connect the 4400 to your LAN. This card can be plugged into the PCMCIA card slot of the 4400 and is fitted with a plug for an Ethernet connection (10 Base T).

> **NOTE**
>
> The 4400 supports only a limited number of different Ethernet card types. It is strongly recommended that you use Willtek's ethernet card.

Option	Willtek order number
4478 TCP/IP Option	M 248 632

Turbo Option (for GSM and GSM/GPRS only)

High throughput is required in mobile production lines and many service factories. Transmitter test speed can be doubled at least with the Turbo Option.

> **NOTE**
>
> This option is not available for the 4400S and the 4403.

Option	Willtek order number
4476 Turbo Option	M 248 359

Test SIMs (for GSM, GSM/GPRS, EDGE and WCDMA)

A Test SIM is a Subscriber Identification Module programmed with a specific mobile phone number, the so-called Test-IMSI (International Mobile Subscriber Identity). A Test SIM is available from Willtek, with the IMSI set to 001-01-0123456789. The 4400 knows this IMSI and uses it to page the mobile under test.
The Test SIM can also be used with other mobile phone testers from Willtek.

> **NOTE**
>
> When a mobile is tested without a SIM card, only emergency calls are possible.

Option	Willtek order number
1100 Test SIM	M 860 188
1102 USIM test SIM card (for WCDMA only)	M 860 173

Mobile-specific RF adapters

The cable connection between a mobile under test and the 4400 requires a mobile-specific connection. Some manufacturers provide a mobile-specific cable or a test adapter. For some mobile phones, Willtek also provides some RF adapters. These RF adapters have a mobile-specific connector on one end and the Willtek RF-click connector on the other one. Please consult with your local dealer concerning available RF adapters.
By using the mobile-specific RF adapters, it is very easy to establish a cable connection between the mobile under test and the 4400. Section Cabling provides you with all the technical background information.

> **NOTE**
>
> To connect any RF adapter to the 4400's N-type RF in/out socket, you will
> need the optional RF-click to N-type adapter cable.

Antennas for use on the 4400

When there is a need to perform quick basic tests on a mobile, an antenna might
be good to have. However, there are a couple of drawbacks using an antenna for
real measurements. Section Cabling provides you with all the technical back-
ground information.
If you need to perform reproducible measurements without having the possibility
of using a RF adapter cable, we recommend to work with the optional RF Shield.
This is a metal box that covers the mobile under test and contains the Antenna
Coupler. This Antenna Coupler carries the mobile under test and provides
antennas for all mobile phone frequency ranges.
The RF Shield keeps radio interference away from the air connection. Together
with a coupling loss table, the achievable measurement quality is as good as
using a cable.
The RF Shield is available with either a TNC-type or N-type connector.

> **NOTE**
>
> The RF shield package contains both the RF Shield and the Antenna Coupler.

Antennas on the
4400

Antenna Coupler

RF Shield

Accessory	Willtek order number
Antenna for GSM 900 with TNC connector	M 860 261
Antenna for GSM 1800/GSM 1900 with TNC connector	M 860 262
4916 Antenna Coupler	M 248 641
4921 RF Shield (N) (including RF cable N – N)	M 248 346
Rear panel for RF Shield customisation	M 300 850
RF Shield Package (4921 RF Shield, 4916 Antenna Coupler)	M 248 349

MS Power Supply Option and Current Measurement Option

The MS Power Supply is an option that simulates the power supply (battery) of the mobile under test. Some of the benefits are:

– It can be used to replace an extra power supply, which may be expensive and consumes space on the desktop or in a rack.

– As with any simulated battery, you can set operating, under or overvoltages in a wide range to check the mobile's reaction to that on the RF side.

In conjunction with the Current Measurement option, there are even more applications feasible to test the mobile phone:

– You can measure the average and peak current the mobile draws from the battery (e.g. when it transmits a burst).

– Even a current vs. time measurement with graphical display can be obtained.

– Last but not least, the average power requirement is calculated from the measurement results.

Option	Willtek order number
MS Power Supply Option	M 248 355
MS Current Measurement Option (requires MS Power Supply Option)	M 248 356

Rack installation, cases

Rack Mount Set

Your 4400 can easily be mounted in a 19 inch rack. The optional Rack Mount Set comprises all the required material including a 19 inch front panel (5 HU) with all necessary cutouts for cables etc.

Option	Willtek order number
4490 Rack Mount Set	M 378 260

Hard cases

The "Carrying Case" for the 4400 is a hard case that is an ideal protection whenever the 4400 needs to be shipped or transported. It also is a safe place to store the test set.
Wheels and handle of the carrying case ease transportation. The telescopic handle can be retracted when not in use.
Color: White, with black wheels and handle.
Dimensions: 700 x 400 x 360 mm (1.040 x 400 x 360 mm with handle fully pulled out).

Option	Willtek order number
Carrying Case	M 300 808

GSM/GPRS Call Mode Operation

2

This chapter describes the functionality of the instrument. Topics discussed in this chapter are as follows:

Overview

This section covers information on GSM and GPRS measurements in synchronous (call) mode. For information on GSM measurements, see "GSM tests" on page 38. There you will find also underlying information on test menus, parameters and outputs in GPRS test mode. Section "GPRS Tests" on page 110 covers information specific to GPRS measurements.

GSM/GPRS mobile phones are typically first tested in standard GSM mode, and then additional (optionally multislot) measurements in GPRS mode are made. The test procedure of the 4463 GSM/GPRS System Option and 4468 EDGE System Option GSM/GPRS System Option is tailored to this common practise. Accordingly, the 4463 GSM/GPRS System Option and 4468 EDGE System Option gives you quick access to GSM test mode via the GSM Basic menu. From the GSM Basic menu, you gain access to the GPRS Basic menu by pressing a softkey.

To select the synchronous test mode, press the **Calls & Meas.** softkey on the Welcome menu. The GSM Basic menu relevant to GSM test mode will appear. To access GPRS test mode, press the **GPRS** softkey on the GSM Basic menu. The GPRS Basic menu will appear.

This system option can test mobile phones designed for the following frequency bands:

– GSM 850

– GSM 900 (P-GSM 900, E-GSM 900, R-GSM 900)

– GSM 1800 (also known as DCS 1800 or PCN)

– GSM 1900 (also known as PCS 1900).

GSM tests

Your 4400 comes with many clever thought-through features that keep GSM testing as simple as possible. There is no need for you to study piles of literature. With the 4400, you will achieve meaningful measurement results with just the push of a few buttons.
Section "Crash course" on page 4 gives a short guideline on how to connect a mobile and perform tests.
Need to know more about the user interface of the 4400? Just navigate to "Menus and levels" on page 20 and you will find all you need to know.

This chapter is divided into three subsections:

"GSM tests" – Here you will find everything regarding the available tests (e.g. Phase/Frequency, MS Info) and how to read the results.

"System parameters" – This subsection will introduce you to the system parameters that are used for the GSM tests. The 4400 comes with a carefully selected set of system parameters that allow you to instantly achieve meaningful test results. However, many parameters can be altered to allow you tailoring the 4400's test capabilities around your needs.

"GSM basics" – If you are not completely familiar with GSM or just need a quick introduction – this is where you will find some explanation of the GSM basics. This chapter also carries many tables on frequency ranges, power classes, burst definitions and so on.

The 4400 provides you with the following test capabilities:

"Basic test menu" – For a quick overall test of a mobile

"Phase/Frequency test menu" – TX tests on phase and frequency behavior

"Burst test menu" – The burst and power/time template measurements

"BER test menu" – The quality of the mobile's receiver, tested using the BER (bit error ratio) measurements

"MS Info test menu" – The mobile's internal data and its SMS (Short Message Service) functionality

To switch from one test to another, simply use the marker tabs on the right-hand side of the screen.

NOTE

If a change of the current test menu is not possible (e.g. because a BS call set up has not been completed), the marker tabs will be grayed out.

On every test menu, there are various functions available like e.g. MS Call, BS Call, Paging, Audio Loop, Start BER, Release Call and so on. These functions can be called by a push on the related softkey below the screen. A more detailed description of the softkeys can be found in section "Softkeys at test level" on page 70.

The burst test menu provides zooming functions that help you to check the mobile's behaviour in detail. For more information on those zoom levels, please refer to section "Burst test menu" on page 47.

Basic test menu   Of course, the main application of this menu is speeding up the test of the mobile's most important characteristics.
Together with the possibility to adapt the main test parameters to specific requirements and the reduced signaling functionality, this basic test menu is an ideal test-time saver.
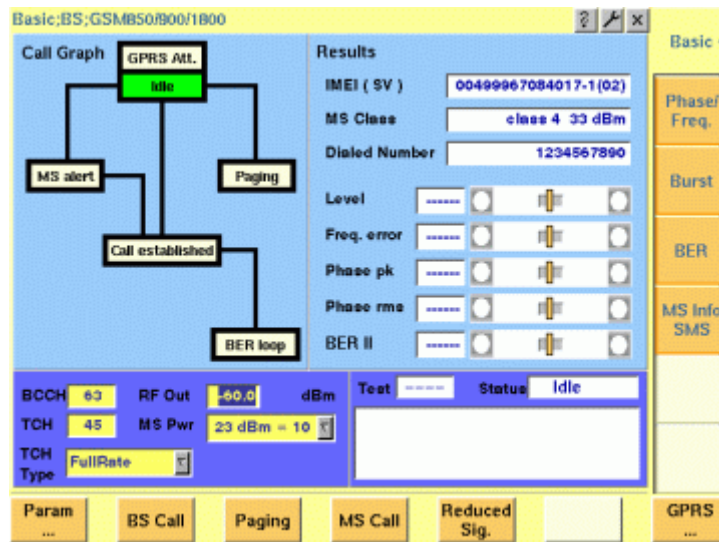
The basic test menu provides you with

– a graphical overview of the ongoing signaling

In addition to information described in "Call graph" on page 41, the call graph provides an additional GPRS Attach field to indicate whether the mobile under test is GPRS attached:

– The GPRS Attach field signals by white background that GPRS data transfer is supported but the mobile is not GPRS attached.

– The GPRS Attach field signals by green background that the mobile is GPRS attached.

> **NOTE**
>
> The GPRS Attach can be performed on both the GSM and the GPRS menus of the 4463 GSM/GPRS System Option and 4468 EDGE System Option GSM/GPRS System Option. The GPRS measurements, however, are supported from the GPRS menus only.

– the most important test results of the mobile's transmitter and receiver in both graphical and numerical formats

– the mobile's IMEI and its power class

– the display of the number dialed on the mobile in case of a mobile-origi- nated call setup

> **NOTE**
>
> Before any RX test results can be displayed, a call has to be set up and the closed loop needs to be established inside the mobile. Please refer to sections "The softkeys" on page 15 and "BER test menu" on page 52 for explanation.

> **NOTE**
>
> A detailed description of the Test and Status area of this menu can be found in section "Status" on page 69.

Call graph   Basically, the call graph is the call status display in graphical form. It allows you to instantly identify the status of a call and the currently executable measurements.

As long as no call or paging has been initiated, the call graph will look like the
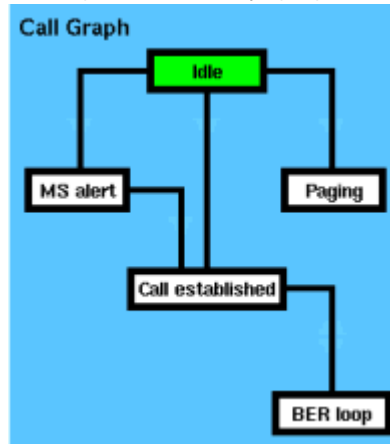
sample above.
In this case, the 4400 is in idle mode and waiting to either receive a call from the mobile or for you pushing either the **BS Call** or the **Paging** softkey.

How to read the call graph

The rectangular box with the green background indicates the current state of the call.

The black lines without arrows indicate the possible paths from one state to another.
Example: There are two paths from Idle to Call established. Either the direct one by a MS-originated call or through the MS alert state during a BS-originated call setup.

All rectangular boxes with white background indicate states that can be reached using the paths.

The orange line with the arrow indicates the path that has been used to achieve the current state.

BS-originated call

To initialize a BS-originated call, simply push the **BS Call** softkey while the 4400 is in idle mode. The test set will then start to establish a call. The call graph will change as follows:
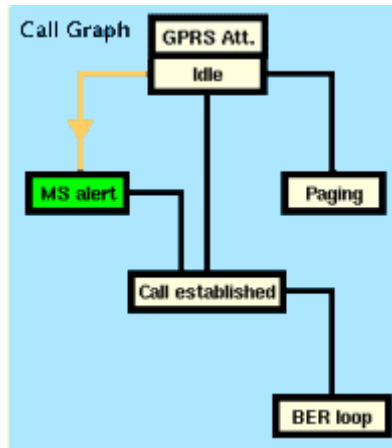
The active status is now MS alert. As soon as all required signaling has been exchanged between the 4400 and the mobile, the status will change to Call established:
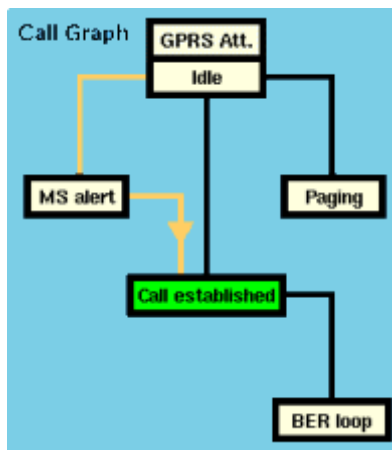
As soon as the call has been put through, you are working on call level. Now, all GSM TX measurements will be performed automatically (for details, see section "Basic test results display" on page 43).

NOTE

If the call can't be established, the 4400 will automatically return to idle mode and the softkeys will change back to their defaults (as described in section "Softkeys at test level" on page 70).

MS-originated call

To initialize a MS-originated call, push the **MS Call** softkey or simply dial a number on the mobile and push the 'send' button. The 4400 will respond to the request of the mobile. The call graph will change to Call established.

As soon as the call has been put through, you are working on call level. Now, all GSM TX measurements will be performed automatically (see section "Basic test results display" on page 43 for details).

BER loop

For the GSM receiver test, the mobile has to enter a specific test mode, called the 'closed loop'. To enter that mode of the mobile, simply push the **Start BER** softkey after a call has been established. The call graph will indicate the successful closing of the mobile's loop as follows:
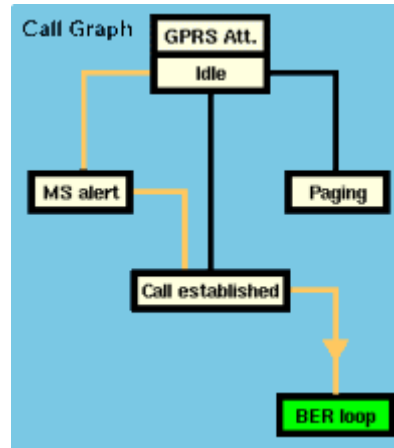
NOTE

This test will only work with a test SIM being used in the mobile.

Paging

The basic test menu also allows you to check the mobile's response to the 4400's paging requests. While working in idle mode, simply push the **Paging** softkey and the 4400 will start broadcasting paging requests. The 4400 will not answer to the mobile's access bursts but will keep on paging it. This allows you to

- check whether the mobile is able to receive the 4400's paging requests on the currently set RF Out level of the test set and to
- repeatedly measure the mobile's access burst.

The call graph will indicate the paging state as follows:
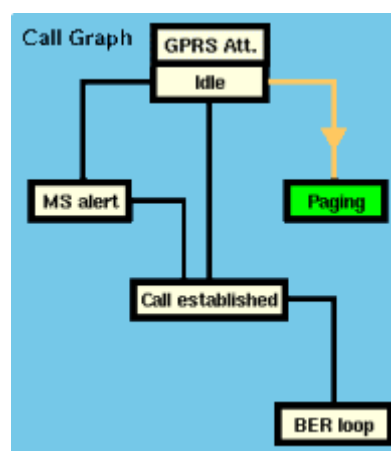
Basic test results display   The basic test results display gives a short overview of the mobile's most important characteristics.

> **NOTE**
>
> Test results will only be shown after a call has been established.

How to read the basic test results display

IMEI (SV)	Here, the 4400 displays the IMEI (= International Mobile Equipment Identity), received from the mobile. The IMEI is the registration number of the mobile hardware which identifies it internationally. The IMEI is stored inside the mobile's electronics and can only be read out by the 4400 through the radio interface. The IMEI consists of 15 digits; the last one is not transferred over the radio interface but calculated from the other 14 digits. The additional two digits shown in brackets indicate the software version in the mobile phone which has been transferred to the 4400 together with the IMEI.
MS Class	This display field shows the mobile's power class both as GSM power class and as the maximum RF output level of the mobile in dBm.
Dialed Number	Following a MS-originated call, the 4400 displays the number that was initially dialed on the mobile. **Note:** This display field will remain empty unless the call has been originated by the mobile.
Level	On this bar display, the 4400 shows the results of the statistic evaluation of the power level measurements. **Note:** The limits for this test are set on the measurement limits menu. This menu is a submenu of the "Basic system parameters" menu.
Freq. error	Here, the 4400 displays the results of the statistic evaluation of the frequency error measurements. **Note:** The limits for this test are set on the "Measurement limits" menu.

Phase peak

The results of the statistic evaluation of the peak phase error measurement are displayed on this bar.

Note: The limits for this test are set on the "Measurement limits" menu.

Phase rms

On this bar display, the 4400 displays the results of the statistic evaluation of the mean phase error (rms).

Note: The limits for this test are set on the "Measurement limits" menu.

BER II

Here you will find the results of the statistic evaluation of that bit-error measurement type selected in the *Mode* selection field of the "BER test menu".

Notes:

– Results will only be displayed after the BER measurement loop of the mobile has been closed by pushing the 4400's **Start BER** softkey.

– The limits for this test are set on the "Measurement limits" menu.

– BER II is used as a synonym for the bit error rate measurement on class II bits (see section "BER live display" on page 53 for technical explanations).

Phase/Frequency test menu

This test menu gives you all relevant parameters required to judge the modulation quality of the mobile. The "Phase/Frequency live display" is updated continuously and provides you with a diagram, showing the phase deviations over the duration of the entire burst.

The statistical evaluation underneath the graphics gives the 'hard facts' and the position of the test results with respect to the measurement limits.
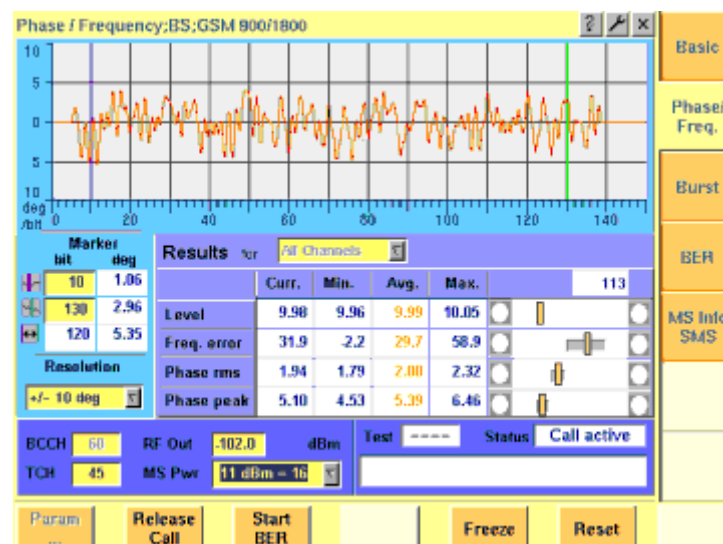
The display markers make it quite easy to read the phase error at any position of the live display.

On first sight

– Level gives the RF power level received by the 4400 (in dBm).

– The Freq. error shows the mobile's carrier offset to the expected frequency (in Hz).

– The Phase rms expresses the mean difference between the modulated signal and the perfect signal (in degrees, rms-valued).

– The Phase peak is the maximum phase error detected within a burst (in degrees).

> **NOTE**
>
> The current MS power level, the mobile has been commanded to use, can be seen and altered in the parameters section of this menu.

> **NOTE**
>
> In this example, there is a difference of some 1 dB between the commanded power level and the currently measured results, which is quite normal. However, that difference could also be due to power losses on connectors and RF cables. Those losses can be compensated using the "Coupling loss" sub menu of the basic parameters section.

> **NOTE**
>
> The meaning of the softkeys is described in section "Softkeys at call level" on page 73.

> **NOTE**
>
> Need to know more about the *Test and Status* area of this menu? All details can be found in section "Status" on page 69.

Phase/Frequency live display

The phase/frequency live display shows the phase deviation of the bits received from the mobile as a function of the bit number.
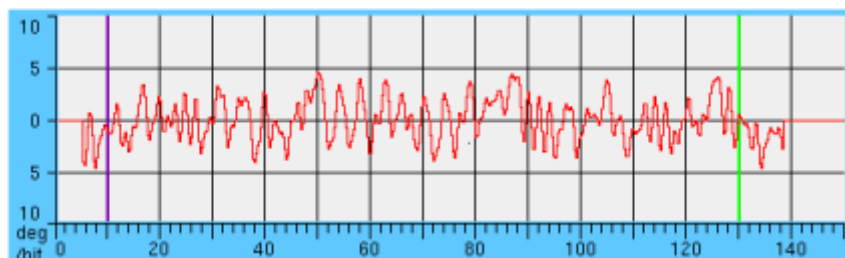
> **NOTE**
>
> To get an exact reading of the values measured, please use the display markers.

> **NOTE**
>
> With the help of the Resolution setting, the vertical axis (the phase error) can be rescaled. The possible settings are: ±2, ±5, ±10, ±20 or ±50 degrees. Default is ±10°.

Phase/Frequency results display   The phase/frequency results display shows the numeric results and the statistical evaluation of the test.

Results for	All Channels					113
	Curr.	Min.	Avg.	Max.		
Level	9.98	9.96	9.99	10.05		
Freq. error	31.9	-2.2	29.7	58.9		
Phase rms	1.94	1.79	2.00	2.32		
Phase peak	5.10	4.53	5.39	6.46		

	113

Shows the total number of burst measurements used to calculate the statistical results.

How to read the phase/frequency test results display

The statistical evaluation of the measurement results is displayed as current, minimum, average and maximum values. The bar displays give a graphical representation of the test results in relation to the measurement limits.

Level   These values give all the RF power level received by the 4400 (in dBm).

Note: The current MS power level, the mobile has been commanded to use, can be seen and altered in the "Test parameters" section of this menu.

Freq. error   The frequency error shows the mobile's carrier offset to the expected frequency (in Hz).

Phase rms   The root-mean square value of the phase error expresses the mean difference between the modulated signal received by the 4400 and the perfect signal (in degrees, rms-valued).

Phase peak   This value gives the maximum phase error measured within a single burst (in degrees).

Burst test menu   The burst test menu allows a number of measurements to determine the mobile's burst quality.
The most important test tool is the power/time template as specified in the relevant GSM standards. On the 4400, the power/time template can be adjusted to your needs using the "Power/time templates (PTT)" menu.
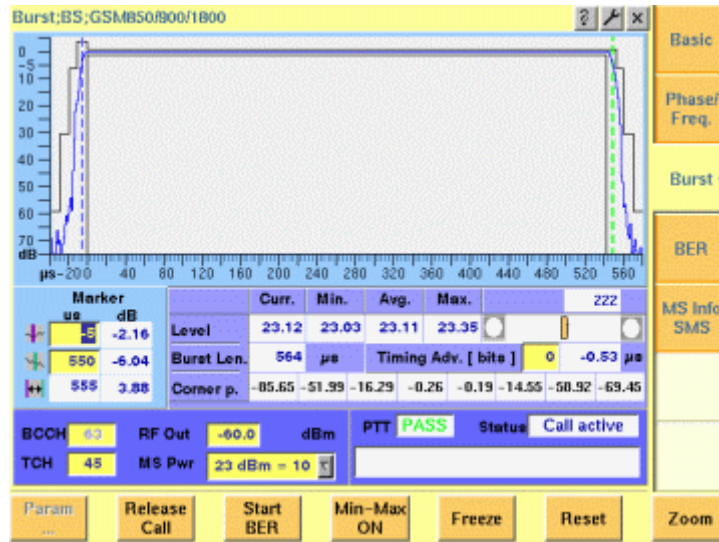
The mobile's burst is displayed graphically in the "Burst live display". The live display is updated continuously. To read the exact measurement at any position, simply use the display markers.

The results of the statistic evaluation of the single measurements are given in the menu's results area.

On first sight

– While the mobile has to ramp up or down its RF power level within some 30 microseconds, the RF power level should be constant during the burst. The **PTT** (= power/time template) defines the power level the mobile should transmit with for any time position during and some 50 microseconds before and after the burst.

– The *Timing Advance* commands the mobile to send its burst a little earlier. Thus, the traveling time of a burst from a mobile far away from the base station does not cause interference with bursts from a mobile close to the base station, transmitting in the subsequent time slot. A more detailed description of the timing advance can be found in section "Time slots and bursts" on page 98.

━━━ **NOTE**

The current MS power level that the mobile has been commanded to use, can be seen and altered in the "Test parameters" section of this menu.

━━━ **NOTE**

The meaning of the softkeys is described in section "Softkeys at call level".

━━━ **NOTE**

Need to know more about the Test and Status area of this menu? All details can be found in section "Status".

Burst live display   The burst live display shows the shape of the burst against the GSM standard or a user-defined power/time template.
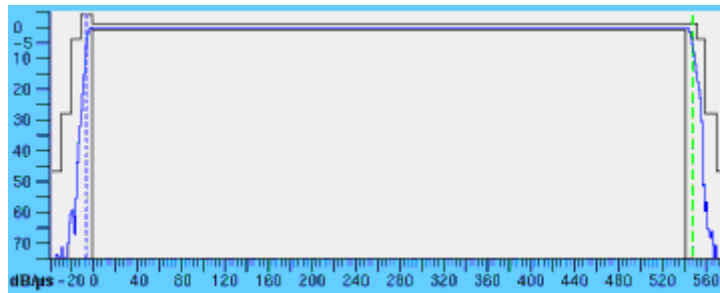
In order to get the exact measurement on any point of the burst, please use the display markers.

The **Zoom** softkey allows you to zoom it the two most important areas of the burst:

– the ramping phase and
– the constant power phase.

Burst results display   The result display shows the numeric results of the burst measurements.
For any GSM network, it is very important that the mobile keeps its RF power level within the limits of the power/time template in order to maintain a good transmission of the data during a burst and not to disturb the transmission of other mobiles during the remaining time slots.

How to read the burst result display

Shows the total number of burst measurements used to calculate the statistical results.

Level

These values give the RF power level received by the 4400 (in dBm); the bar display gives the position of the measured values with respect to the measurement limits.
Note: The current MS power level that the mobile has been commanded to use, can be seen and altered in the parameters section of this menu.

Burst Len.

This display field shows the duration of the burst in microseconds.

Timing Advance This is both an entry field and a display field. On the entry field, you set the timing advance (in bit periods) the mobile has to use. The display field shows the measured timing error (in µs), taking the commanded timing advance into account.

Note: An explanation of the timing advance can be found in section "Time slots and bursts" on page 98.

Corner p. These display fields show the RF power level measured at the corner points. Those corner points are defined (and explained) in the "Power/time templates (PTT)" menu in the Parameters area.

Burst ramp zoom menu This menu shows the magnification of the power up and power down ramping phases of a burst.
The power up ramping phase stretches from 40 microseconds before the burst (-40 µs) to 40 µs after the burst started. The power down ramping phase starts at 500 µs and lasts till 580 µs. Both phases are separated by a thick gray, vertical line.
The resolution of the vertical axis can be set using the selection field Resolution.

NOTE

The PTT limits, as set on the "Power/time templates (PTT)" menu, are shown in gray in this menu.
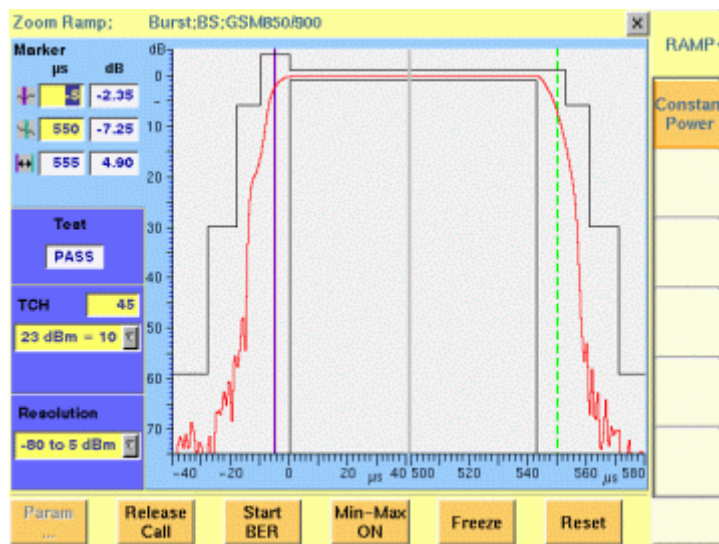
How to read and work with the burst ramp zoom menu

Marker To read the exact measurement values for specific positions, please use the display markers.

Test	Here, the 4400 tells you whether the current burst is ok or whether it has violated any power/time template limits. **Note:** This is important information as the violation could have occurred in the constant power phase of the burst. However, that part of the burst is not visible in this menu.
Test parameters	On this menu, only the TCH channel number and the mobile power level may be altered.
Resolution	With the help of this selection field, you may set the vertical resolution according to your requirements. The possible settings are: –15 to +5 dB, –35 to +5 dB or –75 to +5 dB.
Special functions	Please use the softkeys to call specific test functions.
Constant Power	A simple push on this marker tab changes to the constant power zoom menu.

Burst constant zoom menu   This menu shows the magnification of the constant power phase of the burst. The resolution of the vertical axis is ±5 dB of the expected RF power level of the mobile.

To enhance the horizontal resolution, the graphic display is cut in two parts. The upper one shows the first 280 µs of the burst, while the lower one shows the second 'half' of the burst from 260 to 540 µs.

> **NOTE**
>
> The PTT limits, as set on the "Power/time templates (PTT)" menu, are shown in gray on this menu.
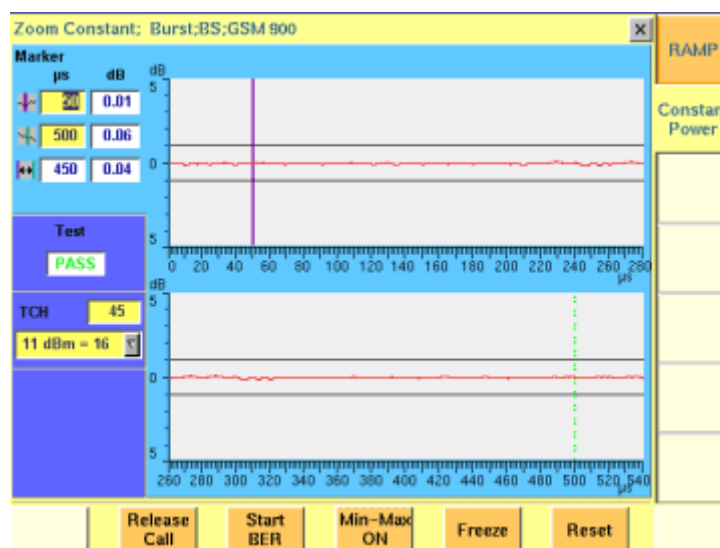
How to read and work with the burst constant zoom menu

Marker	To read the exact values for specific positions, please use the display markers.
Test	Here, the 4400 tells you whether the current burst is ok or whether it has violated any power/time template limits.
	Note: This is important information as the violation could have occurred in the ramping phase of the burst. However, that part of the burst is not visible in this menu.
Test parameters	On this menu, only the TCH channel number and the mobile power level may be altered.
Special functions	Please use the softkeys to call specific test functions.
Ramping Zoom	A simple push on this marker tab changes to the ramping zoom menu.

BER test menu

The BER test menu allows to judge the quality of the mobile's receiver. The results provided are based on a statistical analysis over a number of bursts and bits. The menu area showing the test results is called the "BER live display".
As on all test menus, the most relevant parameters can be altered directly.
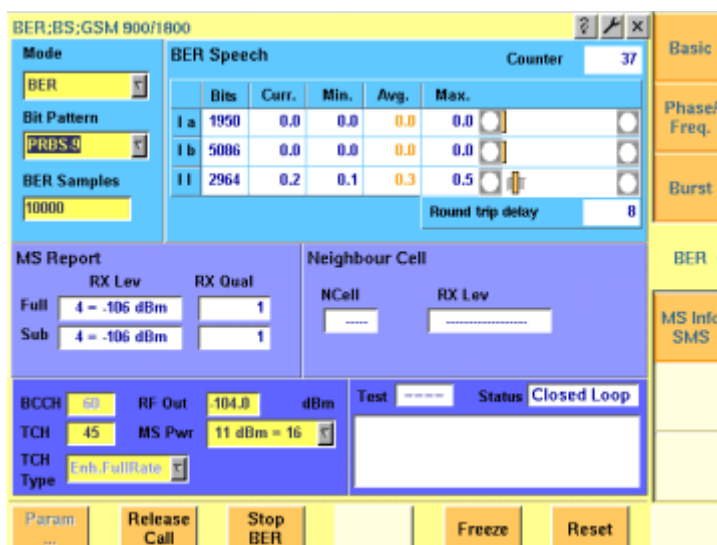
On first sight

– The **bit error rate (BER)** is the ratio of the bits decoded incorrectly by the mobile in relation to all bits received by it.
The BER is tested for different classes of speech and data bits.

– The MS Report is a short form of the data shown on the "MS Info live display".
On the BER test menu, only the parameters informing you about the mobile's own quality interpretation of the bits received will be displayed.

– The mobile also checks the signal quality from **neighbouring cells** and reports those measurements back to the base station (the 4400).

NOTE

The meaning of the softkeys is described in section "Softkeys at call level".

NOTE

Need to know more about the Test and Status area of this menu? All details can be found in section "Status".

BER live display   The three entry fields of the upper left-hand section allow to set specific test parameters. Depending on the settings here, the live display of the test results will vary.

Mode

BER

Bit Pattern

PRBS-9

BER Samples

10000
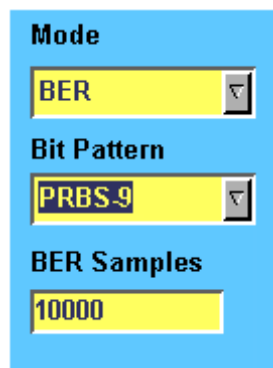
Mode   On this selection field, you may select between BER, Fast BER and RBER measurements.

BER The BER (= **B**it **E**rror **R**ate) is defined as the ratio of the bits that the mobile receives incorrectly to all bits transmitted. The BER is calculated for all three bit classes independently. As you know, bits of the Ia class are highly protected against transmission errors by the so-called channel encoding process. This complicated procedure is used, because any loss of class Ia bits will have a great effect on speech quality.

Class II bits are not protected at all and there is no channel encoding on either side. They are just sent as 'plain vanilla' data from the 4400 to the mobile and back. This is the reason why people concentrate on the BER of class II bits when a quick and simple test of the mobile's receiver is needed: The test data is sent back and forth without any coding and error correction. This allows to solely test the mobile's receiver rather than a combination of the mobile's receiver, decoder and encoder (see also section "Basic test menu" for details).

Note: A more detailed description of the voice coding procedure can be found in section "Voice coding".

FastBER The Fast BER is similar to a BER measurement on the class II bits. The main difference is, that in Fast BER, there are only 'class II bits' and all 456 bits of a voice block are used for the measurement, while there are only 78 class II bits in a standard voice block.

In Fast BER, there is no decoding and encoding on the mobile side. The PRBS received from the 4400 will just be fed back to the mobile's transmitter (PRBS = **P**seudo**R**andom **B**it **S**equence).

Therefore, the Fast BER is a method to solely test the mobile's receiver performance rather than testing the mobile's receiver, decoder and encoder stages.

Fast BER also saves time: To do a standard BER on 10,000 class II bits takes roughly five times longer than doing a Fast BER.

RBER   The RBER is the **R**esidual **B**it **E**rror **R**atio. This is very similar to the standard BER with the following exception:
If the channel decoder of the mobile detects a bad frame, the mobile will ignore the entire frame and signal that back to the 4400 (with a frame containing zeros for the speech block only).
The 4400 will just count how many frames the mobile rejects and will not use the bad frames for calculating the BER. In short words, a RBER is a BER measurement on those frames only that are accepted by the mobile.

Notes:

- A frame is regarded 'bad', when the channel decoder of the mobile detects an error. Usually, the mobile's channel decoder uses the checksum transmitted for the class 1a bits to do so. Consequently, a 'bad frame' has at least one error on the class Ia bits. All frames accepted by the mobile can be regarded as having been received error-free. Therefore, the RBER is calculated for bit classes Ib and II only.

- In theory, all speech frames could be 'bad'. This would then lead to an infinite test duration if the number of samples (see entry field BER Samples) was the criterion to finish a test run. To avoid that problem, the 4400 will always broadcast as many speech frames in RBER testing as the value entered on entry field BER Samples requires to and then display a result. **Example**: 10,000 BER Samples require (INT(10000/210)+1) = 48 frames. (210 is the number of class Ib bits plus the number of class II bits per frame).

- A more detailed description of the class Ia, Ib and II bits can be found in section "Voice coding".

Bit Pattern   On this selection field you set the bit sequence that will be used for the BER measurements.

Notes:

– PRBS stands for PseudoRandom Bit Sequence. The number following it (*9* in this case) indicates the length of the PRBS, which is calculated using the following formula: length = (2 to the power of N) –1. This means that PRBS-9 is a PseudoRandom Bit Sequence with a length of 511 bits. The number of bits used in a voice block is 456. Both figures contain relatively large prime numbers. Therefore, the same bit sequence will only appear at the same position again if the test is run for at least 233,016 BER samples.

– In addition to PRBS-9, also PRBS-15, PRBS-23, all bits zero (0000...), all bits one (1111...), reversals starting with one (1010...) or reversals starting with 0 (0101...) may be selected.

Note: PRBS-9 is the standard setting for GSM BER measurements.

BER Samples   Here you enter the number of bits that shall be used for the current BER measurement. This number is interpreted by the 4400 as the total amount of all class Ia, Ib and II bits. The permissible entry range is from 2,000 to 1,000,000. A good compromise between test time and quality of results is 10,000.

Notes:

– Both a full-rate and a half-rate TCH voice block consist of 260 data bits. 50 of those bits are class Ia bits, 132 are class Ib bits and 78 are class II bits. Due to block coding and convolutional coding, this 260 bit long voice block data enlarges to 456 bits to be transmitted on the air interface.

– A voice block is often also called a speech frame.

– The 10,000 data bits in the screen shot shown above, mean that some 39 speech frames (of 260 bits each) will be transmitted for a standard BER measurement. A new speech frame is transmitted every 20 ms, so the time needed to test the BER over 10,000 data bits is roughly 0.78 seconds (39 * 20 ms).

– The time required by the mobile to do the calculations on the data received usually is in the area of 50 ms. This time works as a delay time (it will take the mobile some 50 ms to transmit the first data back, but then, the stream of data will be continuous). Consequently, it does not add much to the test time.

– More details on speech frames and diagonal interleaving can be found in section "Voice coding".

Standard BER live display   The BER (= Bit Error Rate) is defined as the ratio of the bits that the mobile receives incorrectly to all bits transmitted. The BER is calculated for all three bit classes independently.

BER Speech					Counter	37
	Bits	**Curr.**	**Min.**	**Avg.**	**Max.**	
I a	1950	0.0	0.0	0.0	0.0	
I b	5086	0.0	0.0	0.0	0.0	
II	2964	0.2	0.1	0.3	0.5	
					Round trip delay	8

How to read the standard BER results display

Counter	On entry field BER Samples, you told the 4400 how many bits it shall use for a single BER-test run. On this display field Counter, the 4400 shows how many times a complete BER test has been performed. In short words: The value displayed here is the number of results that have been used to calculate the statistical evaluation.
Bits	These three display fields show the number of bits of the three bit classes Ia, Ib or II that have been used to do a single BER test run.
Curr.	BER of the current test run. Please see section "Statistical test evaluation" for details.
Min.	Smallest BER of all test runs. Please see section "Statistical test evaluation" for details.
Avg.	Average BER of all test runs. Please see section "Statistical test evaluation" for details.
Max.	Largest BER of all test runs. Please see section "Statistical test evaluation" for details.

Round–trip delay

The round-trip delay is the number of TDMA frames 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 on the uplink. According to GSM/ETSI recommendations, the round–trip delay should be less than 26 TDMA frames.

The round-trip delay is a measure of how fast the mobile can receive, decode and encode data.

Notes:

– Using a full-rate TCH, a voice block is spread out on eight TDMA frames. Consequently, the mobile has to gather those eight TDMA frames in order to be able to decode the voice block sent initially. After the voice block has been decoded, the mobile must encode it again and send it back to the 4400. Therefore, the minimum round-trip delay will be 8 TDMA frames on a full-rate TCH.
If there are SACCH frames or idle frames between the single blocks, the typical round-trip delay is 9 or 13 frames.

– A round-trip delay of 26 TDMA frames equals roughly 120 ms.

– More details on speech frames and diagonal interleaving can be found in section "Voice coding".

Fast BER Live Display

The Fast BER is similar to a BER measurement on the class II bits. The main difference is, that in Fast BER, there are only 'class II bits' and all 456 bits of a voice block are used for the measurement, while there are only 78 class II bits in a standard voice block.

Fast BER				Counter		45
Channel	Curr.	Min.	Avg.	Max.		
-------	1.5	1.1	1.4	1.8 ○	🔲 ○	
				Round trip delay		1

How to read the Fast BER results display

Counter

On entry field BER Samples, you told the 4400 how many bits it shall use for a single Fast BER test run.
On this display field Counter, the 4400 shows how many times a complete Fast BER test has been performed.
In short words: The value displayed here is the number of results that are being used to calculate the statistics.

Channel	In the current software release, this display field has no meaning. Therefore, six hyphens ------ are displayed.
Curr.	Fast BER of the current test run. Please see section "Statistical test evaluation" for details.
Min.	Smallest Fast BER of all test runs. Please see section "Statistical test evaluation" for details.
Avg.	Average Fast BER of all test runs. Please see section "Statistical test evaluation" for details.
Max.	Largest Fast BER of all test runs. Please see section "Statistical test evaluation" for details.
Round trip delay	The round-trip delay is the number of TDMA frames between the reception of a particular burst at the mobile and the transmission of the same burst (on the uplink). As there is no coding and encoding required on the mobile's side, just the loop-back of the data received, the typical round-trip delay in Fast BER is one frame.

Residual BER Live Display

The RBER is the **R**esidual **B**it **E**rror **R**atio. This is very similar to the standard BER with the following exception:

If the channel decoder of the mobile detects a bad frame, the mobile will ignore the entire frame and signal that back to the 4400 (with a frame containing zeros for the speech block only).

The 4400 will just count how many frames the mobile rejects and will not use the bad frames for calculating the BER. In short words, a RBER is a BER measurement on those frames only that are accepted by the mobile.

Residual BER					Counter	22
	Bits	**Curr.**	**Min.**	**Avg.**	**Max.**	
FER	48	0.0	0.0	0.0	0.0	
I b	6334	0.0	0.0	0.0	0.0	
I I	3666	1.2	1.0	1.4	1.8	
					Round trip delay	9

How to read the RBER results display

Counter	On entry field BER Samples, you told the 4400 how many bits it shall use for a single RBER-test run. On this display field Counter, the 4400 shows how many times a complete RBER test has been performed. In short words: The value displayed here is the number of results that have been used to calculate the statistics.
Bits	The figure shown in line FER gives the number of voice blocks used for testing. The numbers indicated in lines Ib and II show the total amount of the class Ib and class II bits that were used to calculate the RBER test results.

Curr.	FER and RBER of the current test run (FER = Frame Erasure Ratio, the relation of the number of rejected speech frames to all frames used for testing). Please see section "Statistical test evaluation" for details.
Min.	Smallest FER and RBER of all test runs. Please see section "Statistical test evaluation" for details.
Avg.	Average FER and RBER of all test runs. Please see section "Statistical test evaluation" for details.
Max.	Largest FER and RBER of all test runs. Please see section "Statistical test evaluation" for details.
Round trip delay	The round-trip delay is the number of TDMA frames 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 on the uplink. According to GSM/ETSI recommendations, the round-trip delay should be less than 26 TDMA frames. The round-trip delay is a measure of how fast the mobile can receive, decode and encode data.

Notes:

– Using a full-rate TCH, a voice block is spread out on eight TDMA frames. Consequently, the mobile has to gather those eight TDMA frames in order to be able to decode the initially sent voice block. After the voice block has been decoded, the mobile must encode it again and send it back to the 4400. Therefore, the minimum round-trip delay will be 8 TDMA frames on a full-rate TCH.
If there are SACCH frames or idle frames between the single blocks, the typical round-trip delay is 9 or 13 frames.

– A round-trip delay of 26 TDMA frames equals roughly 120 ms.

– More details on speech frames and diagonal interleaving can be found in section "Voice coding".

MS Report on BER test menu The mobile sends a measurement report to the base station on a regular basis. This measurement report contains the mobile's estimation of

– the RF power level, at which it receives the base station

– the quality of the signals received from the base station.

For more details on the MS report, please check with section "MS Report display".

How to read the MS Report display

RX Lev	The mobile estimates the RF power level at which it receives the base station (the 4400). The result of those estimations is displayed here.
RX Qual	Based on the number of bits corrected by its own channel decoder, the mobile estimates the BER. For more details on this parameter and how it is coded, please check with section "MS Report display".
Full	The above measurements are performed on the assigned TCH. The measurement values given in line Full are always carried out over a complete SACCH reporting period. **Note:** Should a base station perform e.g. DTX, those values are probably averaged.
Sub	When the base station e.g. performs DTX, the measurement will be carried out only using the number of frames actually received by the mobile. This will be a **sub**set of frames of the entire SACCH reporting period. **Notes:** – As long as DTX is not used, the values given in lines Full and Sub are identical. – For more information on this parameter, please check with section "MS Report display".

How to read the Neighbour Cell display

A neighbouring cell usually is an additional base station providing a base channel that is received by the mobile. While working with the 4400, this would e.g. be a 'real' base station or a second GSM signal generator.

NCell	On this display field, the 4400 shows the channel number of the neighbouring cell the mobile receives best. For more details, please refer to section "MS Report display".
RX Lev	The mobile estimates the RF power level at which it receives the neighbouring cell. The result of those RF power level estimations is displayed here. **Note:** The channel number the neighbouring cell is transmitting on, can be read on the NCell display field.

MS Info test menu This menu consists of two main sections:

– the "MS Info live display", where the most important features of the mobile are displayed and

– the "MS Report display", where the mobile's measurement report is shown.
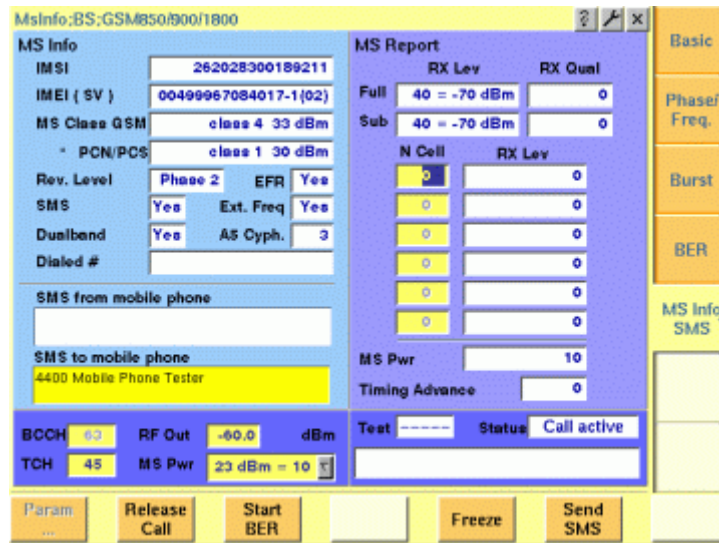
Notes

– The mobile sends a measurement report to the base station every SACCH reporting period (= 104 TDMA frames). In that measurement report, the mobile informs the base station at what power level and with what signal quality it can receive the base station.
Furthermore, the measurement report also gives the channel numbers of neighbouring cells the mobile can receive and the RF power level they come in at the mobile.

– Based on the measurement report, a base station can decide to which neighbouring cell it should hand over the mobile if the radio connection gets worse.

– A SACCH reporting period is approx. 480 ms long.

– The current MS power level, the mobile has been commanded to use, can be seen and altered in the parameters section of this menu.

– The meaning of the softkeys is described in section "Softkeys at call level".

– Need to know more about the Test and Status area of this menu? All details can be found in section "Status".

MS Info live display   This area of the MS Info menu shows the contents of the MS Info as sent by the mobile.

MS Info

IMSI	262028300189211
IMEI (SV)	00499967084017-1(02)
MS Class GSM	class 4 33 dBm
" PCN/PCS	class 1 30 dBm
Rev. Level	Phase 2	EFR	Yes
SMS	Yes	Ext. Freq	Yes
Dualband	Yes	A5 Cyph.	3
Dialed #	

SMS from mobile phone

SMS to mobile phone

4400 Mobile Phone Tester

IMSI International Mobile Subscriber Identity, i.e. the individual identity number of the subscriber in the network. The mobile reads the IMSI from the SIM card.

Notes:
- The IMSI consists of up to 15 digits.
- The first three digits are called the MCC and give the code of the country in which the mobile subscriber is registered.
- The next two digits (the so-called MNC) are the code of the national network the mobile subscriber is registered in. Please note that there are three digits being used for the MNC in North America.
- The remaining ten digits (nine in North America) are the registration number of the subscriber within the network.
- Fore more details on MCC and MNC, please refer to section "Coding of MCC and MNC" on page 103.

IMEI (SV) International Mobile Equipment Identity, i.e. the electronic "serial number" of the mobile. Encoded in the IMEI are: manufacturer, country of origin, place of type approval and the producer's own serial number.
The last digit of the 15-digit IMEI is a check digit that is calculated from the first 14 digits; the result should be the same as the IMEI that the mobile phone displays after entering ***#06#**.
The value in brackets is the software version as indicated by the mobile phone over the air interface.

MS Class GSM	This is the mobile's power class as long as it is working in a GSM 900 environment. The power class gives the maximum RF output power a mobile is able to transmit on. The coding can be found in section "MS power classes" on page 108.
MS Class PCN/PCS	This is the mobile's power class as long as it is working in a PCN (GSM 1800) or PCS (GSM 1900) environment. The power class gives the maximum RF output power a mobile is able to transmit on. The coding can be found in section "MS power classes" on page 108.
Rev. Level	The value shown here indicates whether the mobile supports all features of GSM phase 1 or phase 2 or any later phase.
EFR	EFR stands for Enhanced Full Rate. A Yes indicates that the mobile is able to support enhanced-full rate channels, while a No tells you that the mobile can't. **Note:** This information is only available after a call has been set up, while all other parameters shown on this menu are provided by the mobile during a location update.
SMS	Indicates the mobile's ability to handle Short Message Service functions. A Yes indicates that the mobile supports SMS.
Ext. Freq	Here, the 4400 informs you whether the mobile is able to use the extended frequency range (E-GSM).
Dualband	A *Yes* on this display field indicates that the mobile is able to work in both GSM and PCN (=GSM 1800) networks. **Note:** In order to perform crossband channel assignments, the mobile has to show a Yes here.
A5 Cyph.	Here, the 4400 displays the A5 cyphering key supported by the mobile.
Dialed #	In case the call was originated by the mobile, the number dialed on the mobile will be shown here.
SMS from mobile phone	Any SMS entered on the mobile phone will be displayed here. **Note:** In order to work with SMS, the mobile, of course, has to support that feature (see display field SMS on this menu).

SMS to mobile phone Enter any SMS message on this entry field. To send it to the mobile, just push the **Send SMS** softkey.

Notes:

– If you are working without external keyboard, please refer to section "Entering values on alphanumeric entry fields" on page 24.

– A typical example of a SMS is the following text: "the quick brown fox jumps over the lazy dog" as it contains all letters of the English/ American alphabet.

– In case you leave this entry field empty and push the **Send SMS** softkey, the 4400 will send a standard SMS to the mobile.

MS Report display This section of the MS Info menu shows the full MS Report. On the "BER test menu", a fraction of that information is displayed.

MS Report

	RX Lev	RX Qual
Full	49 = -61 dBm	0
Sub	49 = -61 dBm	0

N Cell	RX Lev
0	0
0	0
0	0
0	0
0	0
0	0

MS Pwr	16
Timing Advance	0

During a call, the mobile sends a measurement report to the base station every SACCH reporting period (= 104 TDMA frames). In that measurement report, the mobile informs the base station at what power level and with what signal quality it can receive the base station's bursts. Furthermore, the measurement report also gives the number of neighbouring cells the mobile can receive and the RF power level they come in at the mobile.

Notes:

– Based on the measurement report, the base station can decide to which neighbouring cell it should hand over the mobile if the radio connection gets worse.

– An SACCH reporting period is approx. 480 ms long.

How to read the MS Report display

RX Lev   The mobile estimates the RF power level at which it receives the base station (the 4400). The result of those estimations is displayed here.

RX Qual   Based on the number of bits corrected by its own channel decoder, the mobile estimates the BER. The coding of the number displayed here can be found in section "Coding of RX Qual" on page 109.

Full   The above measurements are performed on the assigned TCH. The measurement values given in line Full are always carried out over a complete SACCH reporting period.

Note: Should a base station perform e.g. DTX, those values are probably averaged.

Sub   When the base station e.g. performs DTX, the measurement will be carried out only using the number of frames actually received by the mobile. This will be a **sub**set of frames of the entire SACCH reporting period.

Notes:

– Usually, a base station will perform DTX while the mobile's user is talking. During that period, the person on the 'other end' of the line is mainly listening.

– As long as there is no DTX, the values given in lines Full and Sub are identical.

– In case of DTX, the values given in lines Full and Sub might differ: During DTX, a base station only transmits SID frames. Thus, the base station reduces unnecessary interference with other base stations.

NCell   In standard GSM, the base station informs the mobile about the channel numbers of a set of base stations of the same network that are in the geographical area of that base station.

In its 'spare time', the mobile will listen to those base stations and estimate the RF level they come in at the mobile. The channel numbers of up to six base stations can be entered here. In its measurement report, the mobile will report back the estimated RF power levels for those six base stations on the corresponding RX Lev display fields.

Notes:

– The set of neighbouring base stations is sent by the base station in the system-information message on the BCCH (in idle mode) or on the SACCH (during an established call).

– The results of the mobile's measurement report is used to initiate a handover of the mobile to the base station supporting the mobile best.

RX Lev	The mobile estimates the RF power level at which it receives the up to six base stations. The frequency channels on which those neighbouring cells provide their base channels, can be entered on the NCell entry fields.
MS Pwr	The value displayed here is the so-called power control level. The base station tells the mobile what power level to use for its bursts (see section "Test parameters" on page 68 for details). The value displayed here is the confirmation from the mobile on what power level it actually transmits. The figures transmitted here are coded. The power level related to the code displayed can be found in section "Power control levels and tolerances" on page 103.
Timing Advance	On this display field, the 4400 gives the timing advance of the mobile in bit periods. A technical explanation of the timing advance can be found in section "Time slots and bursts" on page 98.

Display markers The 4400 offers two markers for easy reading of the graphic displays...

Marker		
	bit	**deg**
↓∿	10	-1.00
∿↘	130	0.60
↔	120	1.60

...of the phase/frequency test menu

On the Bit entry fields, you set the markers to the required horizontal positions (= bit values). The corresponding result fields (deg) show the measurement values obtained for the two positions in degrees. The third row shows the difference between the selected bit positions and the difference between the two corresponding measurement values.

Marker		
	µs	**dB**
↓∿	-5	-2.84
∿↘	550	-8.69
↔	555	5.85

...of the burst test menu

On the µs entry fields, you set the markers to the required horizontal positions (= time positions, relative to the beginning of the burst). The corresponding result fields (dB) show the measurement values obtained for the two positions in dB.
The third row shows the difference between the selected time positions and the difference between the two corresponding measurement values.

Resolution
+/- 10 deg ▽

By changing the value of the Resolution selection field, you can easily modify the vertical resolution of the graphic display.
Smaller values provide a higher resolution and thus more details; higher values provide a better overview.

Note: This selection field is available in the phase/ frequency test menu only.

Test parameters

These entry fields offer the possibility to easily alter the following important test parameters.

NOTE

All test parameters as well as the limits for the various tests or the power/ time templates etc. can be modified using the "Basic system parameters" menus.

BCCH — Channel number for the base channel. This is the frequency at which the 4400 emits its BCCH, SCH, FCCH and other logical channels (BCCH = Base Station Control Channel, SCH = Synchronization Channel, FCCH = Frequency Correction Channel).

NOTE

The term base channel is used throughout this manual. The GSM specifications use the term 'frequency channel C0 of the cell allocation' or 'BCCH carrier'.

RF Out — RF output level of the 4400's base channel.

TCH — Channel number of the TCH (= Traffic Channel). The TCH contains the speech or data transmitted during a call connection.
If you change the setting of this entry field during a call, a channel assignment to the new channel will be performed automatically.

NOTE

Section "Basic specifications of GSM bands" on page 97 gives an overview of the channel numbers used in the various GSM systems. With the help of the information provided there, it is also possible to calculate the RF frequency from the channel number.

MS Pwr — Power control level. This is the RF output power level the mobile has to use for its TCH.

NOTE

The value entered here is sent to the mobile using standard signaling. As long as the value is within the mobile's RF power capabilities, it has to transmit on that power level.

NOTE

A table of the power control levels can be found in section "Power control levels and tolerances" on page 103.

▬▬▬ | **NOTE**

During a cross-band handover, the 4400 will command the mobile to use the power control level that was used last in the new frequency range.
This offers the possibility to put 'additional stress' on the mobile.
Example: While the mobile is commanded to transmit with the minimum power level in the previous band, it might be commanded to transmit at maximum power level in the newly assigned frequency band – or vice versa.

TCH Type — This selection field determines the type of traffic channel used during a test.
Available types are HalfRate, FullRate and Enh.FullRate.

▬▬▬ | **NOTE**

This selection field is only available in the Basic and BER test menus.

▬▬▬ | **NOTE**

HalfRate is supported by many types of mobiles available. However, the network operators are reluctant to use this feature as they are mainly afraid of loosing speech quality.

▬▬▬ | **NOTE**

FullRate is what GSM started with and what is still the default Codec type.

▬▬▬ | **NOTE**

Enh.FullRate (= enhanced full rate) is a new Codec algorithm providing better speech quality than Full Rate.

Status

The status area shows the overall results of the current test and the signaling status.

Test

PASS	All measurement results of the current test are within their limits.
FAIL	This mark indicates that test limits have been violated by the measurement results.
---- (Four hyphens)	Will be displayed when the measurement hasn't been started yet or when there are no limits available for the current test.

Status

Signaling status from the 4400's point of view. The large display field below gives more details and a short signaling history.

Idle	This is the status as long as there is no signaling active.
Call active	Indicates that there has been a call established.
Paging	This will be displayed as long as the 4400 pages the mobile.
Closed Loop	Will be displayed as long as there is a closed loop established on the mobile's side for e.g. the BER test.
‒ ‒ ‒ ‒ ‒ ‒ **(six hyphens)**	This indicates that the call setup hasn't been started yet or that there has been no communication between the 4400 and the mobile so far.

Softkeys at test level

As long as no call has been set up and no paging started, you are working in test level. In test level, the 4400 displays the following softkeys.

Param ...	Calls the "Basic system parameters" menu on-screen. With the help of this menu, you may set and modify test parameters, limits, the power/time template and so on.
BS Call	A push on this softkey will make the base station call the mobile ("BS Call"). The softkey will change to **STOP**. As soon as the call has been completed, the entire line of softkey labels will change to provide access to additional functions. A description of the softkeys displayed on call level can be found in section "Softkeys at call level" on page 73.
Paging	This softkey will make the 4400 page the mobile under test. As soon as the paging is in progress, the label will change to **Stop Paging** to allow termination. **Note:** The 4400 will not answer to the mobile's access bursts but will keep on paging it. This allows you to: – check whether the mobile is able to receive the 4400's paging requests on the currently set RF Out power level of the test set and to – repeatedly measure the mobile's access burst.
MS Call	Starts a MS-originated call ("MS Call"). You will be prompted to dial a number on the mobile and to push the 'send' button.
Reduced Sig.	After a push on this softkey, the 4400 will start to provide a base channel with reduced signaling. **Note:** More details on this feature can be found in section "Reduced signaling" on page 74.

Send SMS

Sends a SMS to the mobile under test.

Note: This softkey is available in the "MS Info test menu" only.

Zoom

The **Zoom** softkey gives access to the zoom function of the related test menu.

Note: This softkey is available in the "Burst test menu" only.

Marker tabs at burst zoom level

There are two different menus for the burst zoom function: the zoom display of the power ramping phase and the zoom display of the constant power phase of the burst. A push on one of the two marker tabs selects the menu you want to work with.

RAMP

This menu offers you specific display functions to evaluate both the rising and the falling edge of the burst in great detail.

Constant Power

Here you will find all functionalities to check all details of that part of the burst that should have a constant power level.

Statistical test evaluation

Results for All Channels	Curr.	Min.	Avg.	Max.		113	
Level	9.98	9.96	9.99	10.05			
Freq. error	31.9	-2.2	29.7	58.9			
Phase rms	1.94	1.79	2.00	2.32			
Phase peak	5.10	4.53	5.39	6.46			

On quite a number of test menus, the 4400 provides you with statistic evaluations of the measured values. Those statistic values are obtained by running a test a specific number of times. In the figure above, this number is given by the single value above the bar displays (113 in this example).
The following information can be found in the results table:

Curr.   Current measurement value. The last measurement obtained without any statistical evaluation.

Min.   The absolut minimum of all measurements obtained.

Avg.   The average value calculated using all measurement values since the last start of the measurement.

Max.   The absolut maximum of all measurements obtained.

> **NOTE**
>
> The bar display is described in section "Display markers" on page 67.

> **NOTE**
>
> To (re-)initialize a statistic evaluation, simply push the **Reset** softkey.

MS Call

MS Call

A push on this softkey will initiate the MS call procedure.

Information
Please dial a number (Abort with ESC)
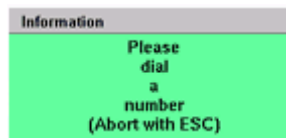
The 4400 will prompt you to dial any number on the mobile and to push the 'send' button.
To abort the call setup, push the **Esc** key.

> **NOTE**
>
> When the 4400 is in idle mode, you may always dial a number and push the 'send' button on the mobile to start a mobile-originated call without having to push the **MS Call** softkey first.

> **NOTE**
>
> While the call is being established, the "Status" section of the menu displays the corresponding status information

> **NOTE**
>
> After the call has been set up, you can perform any measurement, or call specific functions with the help of the softkeys. See section "Softkeys at call level" on page 73 for details.

BS Call
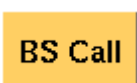
BS Call

A push on this softkey initiates a BS-originated call setup. The 4400 calls the number of the mobile (stored on the Willtek test SIM). The label of the softkey changes to **STOP**.

Notes

– The base station will always call the number of the test SIM card (001-01-0123456789) first.
– The second number the base station will call is the number of the older SIM cards (001-01-1234567890).
– In case a call has already been set up with a mobile using a different number (from a different SIM card), that number will then be used as the second number. The older SIM cards will not be called in that case.

STOP   Cancels the setup of a BS-originated call.

NOTE

In case the mobile does not answer the BS calls, it may not be registered to the network simulated by the 4400.
Please check the settings on the "Basic system parameters" menu, especially the entry fields MCC, MNC and Cell Identity. Please also check that the parameter Cell Barred Access on the same menu is set to '0'.

NOTE

While the call is being established, the "Status" section of the menu gives the corresponding status information.

NOTE

After the call has been set up, you may perform any measurement, or call specific functions with the help of the softkeys. See section for details.

Softkeys at call level

After a call has been set up, you are working in call level. In call level, the 4400 displays the following softkeys.

Param ...   Calls the "Basic system parameters" menu on-screen. With the help of this menu, you may set and modify test parameters, limits, the power/time template and so on.

Release Call   A push on this softkey terminates the call.

Start BER   Tells the mobile to close the internal loop and starts the bit-error rate measurement (BER). At the same time, the 4400 opens the audio loop.
The softkey changes to **Stop BER**.

Stop BER   Stops the BER measurement and tells the mobile to open the internal loop. At the same time, the 4400 closes its audio loop.
The softkey changes back to **Start BER**.

Min-Max ON

Starts the concurrent display of minimum and maximum values. This is like a peak-hold and a minimum-hold function; for each point on the horizontal axis, the smallest and the highest measurement values since the start of the test are shown together with the most recent measurement curve. The most recent measurement curve is shown in blue while the minimum and maximum curves are displayed in magenta, see picture below.
Sampling minima and maxima can be restarted by pushing the **Reset** key. The minima and maxima display can be switched off by pushing the softkey again.

Min-Max OFF

This softkey is displayed when the minima/maxima display function is switched on. It can be used to switch the minima and maxima display function off again.

Freeze

Pushing this softkey freezes the live display. This means, that the measurements taken last will be kept on display and no new measurements will be shown. The label of the softkey changes to **Unfreeze**. Pushing the softkey again reactivates the live display.
Note: The communication link between the 4400 and the mobile is being maintained.

Reset

This softkey (re-)initializes the statistical evaluation of the measurements.
All minimum, average and maximum values are discarded and the measurement started again.

Send SMS

Sends a SMS to the mobile under test.
Note: This softkey is available in the "MS Info test menu" only.

Zoom

The **Zoom** softkey gives access to the zoom function of the related test menu.
Note: This softkey is only available in the burst test menu.

Example of the minima and maxima display (spectrum case):
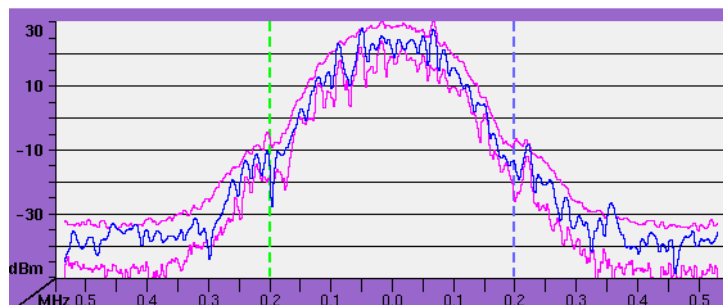
Reduced signaling   The main application for reduced signaling is that it speeds up testing.

Reduced Signaling means that no FACCH (= **F**ast **A**ssociated **C**ontrol **Ch**annel) is being used. All signaling is done on the SACCH (= **S**low **A**ssociated **C**ontrol **Ch**annel).

In order to test multiple mobiles as quickly as possible, the following reduced signaling functionality can be used with the Willtek 4400:

– The 4400 generates a base channel and a traffic channel on different time slots (as in standard GSM).

– While the mobiles are synchronized to the base channel, they should not register.

– One of the synchronized mobiles is commanded to the traffic channel (e.g. through remote control of the mobile) and starts transmission/reception on the assigned traffic channel.

– All tests performed with reduced signaling are 100% identical to the standard testing of mobiles.

– The current TCH an be changed by (remote) control of both the 4400 and mobile.

– Basically, all activities are a combination of remote control of the mobile and adequate settings of the 4400.

Testing with reduced signaling is started by pushing the **Reduced Sig.** softkey or by the SCPI remote command `CALL:GSM:RSIGnaling`.
As soon as you push the **Release Call** softkey, the reduced signaling mode will be terminated.

System parameters

These menus allow to set a number of system parameters, to define the power/time templates for both random-access bursts and normal bursts, to set test limits or to define a coupling loss curve.

To gain access to these menus, simply push the **Param...** softkey while working on the test level of any GSM test.

With the help of the marker tabs, you may select between the following menus:

– "Basic system parameters" on page 76 – in the basic system parameter menu, configuration data of the base station can be set in order to check the behavior of the mobile.

– "Power/time templates (PTT)" on page 81 – allows to set the limit values of the PTT for both random-access bursts and normal bursts for the various GSM bands.

– "Measurement limits" on page 85 – this is the menu to enter the various test limits (except for power measurements). These limits define the acceptable range of measurement results. As long as all results are within the acceptable range, a test will be marked as PASSed.
Any measurement result violating the acceptable range entered here will cause the entire test to FAIL.

– "Power measurement limits" on page 88 – in this menu you can enter limits for the different power steps, separated by bands.

– "Coupling loss" on page 90 – allows to compensate the losses of cables, splitters and other RF equipment. This is especially important when working with the universal antenna coupler.

– "Corner points" on page 94 – with the help of this menu, you define the range of permissible measurement results for eight critical positions of a burst. Those positions can be defined according to your requirements. Especially in combination with the remote control of the 4400, this menu is a powerful and time-saving tool.

Basic system parameters

The basic system parameters menu allows to set a number of parameters of the base station simulated by the 4400.
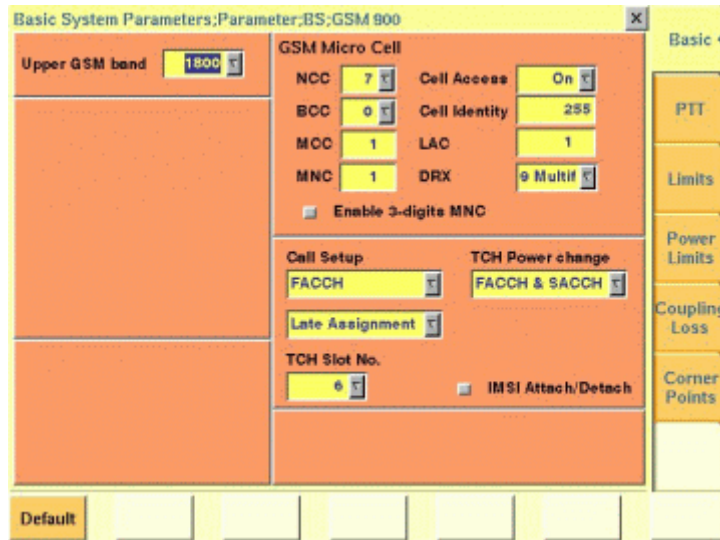
Upper band selection

| Upper GSM Band | A GSM frequency channel can be selected by its channel number. For GSM 1800 and GSM 1900, however, channel numbers are the same. Therefore, the 4400 offers you the possibility to select which frequency band is meant by a channel number of 512 or above. |

**Upper
GSM
Band**

A GSM frequency channel can be selected by its channel number. For GSM 1800 and GSM 1900, however, channel numbers are the same. Therefore, the 4400 offers you the possibility to select which frequency band is meant by a channel number of 512 or above.
Possible entries: `GSM 1800` or `GSM 1900`.
Default: `GSM 1800`

**The parameters of the GSM
Micro Cell area**

NCC

The **N**etwork **C**olour **C**ode is transmitted by the base station on its synchronization channel SCH. The NCC allows the mobile to identify base stations belonging to different (national) networks.
Note: While scanning the frequency range for suitable base stations, the mobile will first look at the base station's FCCH (frequency correction channel) and then at the SCH.
Entry range: 0...7
Default value: 2

BCC

The **B**ase station **C**olour **C**ode makes it easy for the mobile to clearly separate between several neighbouring base stations. Furthermore, the BCC is identical with the training-sequence code.
Entry range: 0...7
Default value: 0

MCC

The **M**obile **C**ountry **C**ode identifies the country the mobile is registered in.
The MCC is coded according to an international standard (CCITT Rec. E .212, Annex A) and is unique for every country. Examples of GSM MCCs can be found in section "Coding of MCC and MNC" on page 103.
Entry range: 0...1000
Default value: 1

MNC

The **M**obile **N**etwork **C**ode indicates the national network, the mobile is registered in.
The MNC is coded by national authorities, usually starting from 01.
Examples of GSM MNCs can be found in section "Coding of MCC and MNC" on page 103.
Note: The MNC is a two-digit code. Exception from the rule: US networks use three-digit coding.
Entry range: 0...100
Default value: 1

Cell Barred Access

When this entry field shows a 0, the mobile will be granted access to the simulated base station. A 1 here will block the base station for access.
Entry range: 0 or 1
Default value: 0

Cell Identity

This parameter gives the number of the cell within the network.
Entry range: 0...255
Default value: 255

LAC

The **L**ocation **A**rea **C**ode is the number of the base station within the network.
Note: Several base stations together may form a cell.
Entry range: 0...65535
Default value: 1

DRX   Discontinuous Reception tells the mobile not to listen to the paging channel in every multiframe but only e.g. to every second or third. The mobile is assigned to one of several 'paging groups', depending on its IMSI. Paging groups allow the network to handle a higher amount of mobiles within a cell, while the mobiles save battery power. The number of multiframes over which the paging groups are spread is broadcast with the BS_PA_MFRMS parameter in the System Information messages. The BS_PA_MFRMS is equivalent to the DRX parameter and allows spreading over 2 to 9 multiframes.
Note: A multiframe is a period of 51 TDMA frames and equals roughly 235 ms in time.
Entry range: 2 Multif ... 9 Multif
Default value: 2 Multif

Enable 3-digit MNC   If checked, the 4400 allows you to enter any three-digit MNC.
Usually, the Mobile Network Code is comprised of two digits (as defined in the ETSI standard). In North America, however, a three-digit code is used with a sligthly modified protocol. This modified protocol is used if you check this box.
Entry range: On or Off
Default value: Off

The parameters of the Call Setup area

Call Setup   There are two standard signaling channels in GSM: the relatively slow SDCCH and the fast FACCH (SDCCH = **S**tand-alone **D**edicated **C**ontrol **Ch**annel, FACCH = **F**ast **A**ssociated **C**ontrol **Ch**annel).
The FACCH is always associated with the TCH (TCH=**T**raffic **Ch**annel). As the test of a mobile needs a TCH to be established anyway, the FACCH will be used for signaling as a standard. However, some mobiles will listen to signaling provided on the SDCCH only. This is of course not compliant with the GSM specifications, but a test set needs to be flexible.
Note: In the current software release, the entry on this selection field can't be changed.
Entry range: FACCH or SDCCH
Default value: FACCH

Assignment	GSM knows two different TCH assignment procedures that can be selected here (TCH = Traffic Channel). Early Assignment means that the mobile will be guided on a TCH as soon as possible. In case of a mobile-originated call, this means before the call has been routed completely; in case of a mobile-terminated call, this will be the case as soon as the 4400 receives the 'call confirmed' message from the mobile. Late Assignment is the procedure that guides the mobile on a TCH any time after the first 'alert' message has been sent (i.e. the TCH will be established after the phone is ringing on the other end of the line). **Entry range:** Late Assignment or Early Assignment **Default value:** Late Assignment
TCH Slot No.	On this selection field, you select the time slot the 4400 is to use for the TCH (Traffic Channel). **Note:** The base channel will always be transmitted during time slot 0. Slots 1 and 7 are used by the 4400 to switch to the TCH and back. **Entry range:** 2..6 **Default value:** 2
TCH Power Change	In GSM, the mobile can be commanded to use a certain power level. The command to do so can either be transmitted on the FACCH (**F**ast **A**ssociated **C**ontrol **Ch**annel) or on the SACCH (**S**low **A**ssociated **C**ontrol **Ch**annel). The main difference is timing, of course. Commands transmitted on the FACCH (stealing channel) will be effective extremely quick and thus lead to short measurement times. However, some mobiles don't seem to like that and therefore, the 4400 also provides the SACCH-only way, as the GSM network does. **Entry range:** FACCH&SACCH or SACCH **Default value:** FACCH&SACCH

IMSI Attach/ Detach

This check box is used to select the Attach/Detach procedure. The advantage of this procedure is that you get a visible notification when the mobile is camping on the cell simulated by the 4400.

If this box is selected, the 4400 will initially indicate the Idle indication within the call graph with a white background. The 4400 signals to the mobile phone to apply the IMSI Attach/Detach procedure. This means that when the mobile is turned on and has selected the cell simulated by the 4400, it will register with the 4400. This way, the 4400 can notify the user when the mobile is ready for a call. The indication is given by the background of the Idle box within the call graph, turning from white to green. This way, you will know when the mobile is ready for testing. Similarly, when the mobile is turned off (or when it selects a different cell), it will de-register with the 4400. The background color of the Idle box will subsequently turn back to white, indicating that no mobile is ready for testing.

If the box is deselected, the IMSI Attach/Detach procedure will not be performed; this is the default state and should be used to speed up testing, especially under remote control.

Entry range: On or Off
Default value: Off

Resetting all entry fields of the 4400 to the default values

Default

Warning: A simple push on this softkey will cancel all values entered on this and all other menus and set all of them to factory defaults. All your previous entries will be lost.

Power/time templates (PTT)

In GSM, there are two different types of bursts: the relatively short random access bursts and the usual normal bursts. For both burst types, individual limits for the power/time template can be set using the corresponding PTT menu. The menu shown below is for the random-access bursts.
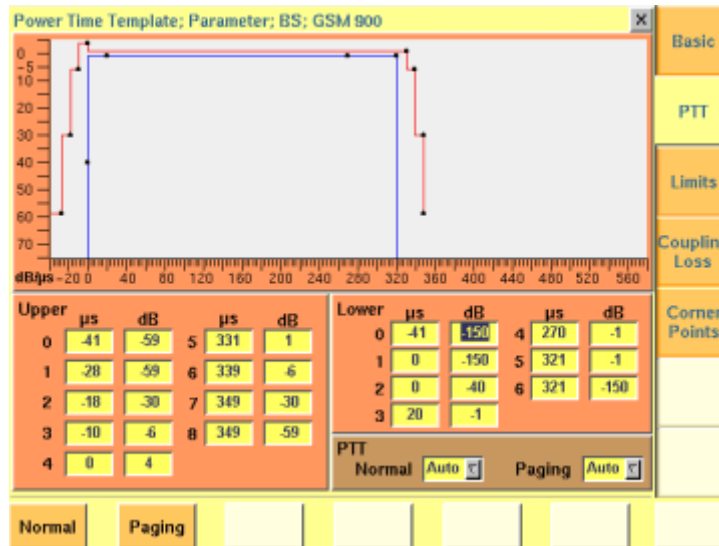
PTTs for normal bursts and random-access bursts

Normal

A push on this softkey switches to the menu for entering the limits of normal bursts. This menu is to be identified easily by the template stretching out for the entire width of the menu.

Paging

Guides you to the menu for entering the limits of random-access bursts. Those bursts are significantly shorter than normal bursts; therefore the display on-screen only covers about half of the width.

Selecting between auto and manual mode

According to the GSM specifications, the limits for the power/time templates vary with the mobile's peak power level allowed by the GSM band you are currently working in. The 4400 comes with a built-in table for all the PTT parameter sets.

Normal
Sets the limits for the normal bursts either to *Auto* mode or to manual mode (Fix).
In Auto mode, the 4400 will always use the PTT limit parameters as given by the relevant GSM specifications. In manual mode (Fix), only the limits entered on the normal-burst PTT menu will be used, regardless of GSM band and the mobile's peak power level.

Note: More details regarding the Auto settings can be found in section "PTT limits for various GSM bands" on page 106.

Entry range: Auto or Fix
Default: Auto

Paging   Sets the limits for the random-access bursts either to *Auto* mode or to manual mode (Fix).

In Auto mode, the 4400 will always use the PTT limit parameters as given by the relevant GSM specifications. In manual mode (Fix), only the limits entered on the random-access burst PTT menu will be used, regardless of GSM band and the mobile's peak power level.

Note: More details regarding the Auto settings can be found in section "PTT limits for various GSM bands" on page 106.

Entry range: Auto or Fix
Default: Auto

Configuring individual limits for the PTTs

A power/time template consists of two limit ranges: the upper range and the lower range.

The upper range is indicated by the red line in the PTT graphics. The values entered for the upper range form the template for the maximum RF power level of the mobile; i.e. the mobile's burst must not exceed those limits at any time. The lower range is shown by the blue line. The values entered for the lower range give the power levels the burst must not fall below.
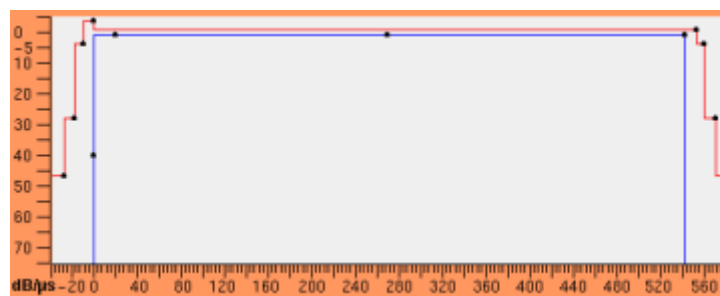
Entering and reading the upper range

> **NOTE**
>
> The entry fields shown below are for normal bursts. The same principles apply for random-access bursts.

> **NOTE**
>
> The time reference (t = 0) is the beginning of the first bit of the 148 bits of the normal burst. For details on a normal GSM burst, please check with section "Time slots and bursts" on page 98.

Upper	µs	dB		µs	dB
0	-42	-47	5	553	1
1	-28	-47	6	561	-4
2	-18	-28	7	571	-28
3	-10	-4	8	580	-47
4	0	4			

The upper range of the power/time template is made up by a number of value pairs. Those value pairs give a position in time relative to the beginning of the burst and a power level. Plotting these value pairs and connecting them with lines will form the red graph of the PTT.

Example: In the screen shot above, the following two value pairs are given:
4: 0 µs, 4 dB
5: 553 µs, 1 dB
This means: from 0 to 553 µs, the power transmitted by the mobile must not exceed its nominal output power level by more than 1 dB.
From 10 microseconds before the burst (value given in line 3) until the burst starts, the mobile must not exceed its nominal output power level by more than 4 dB.
From 553 to 561 µs (latter value given in line 6), the measured power level must be at least 4 dB lower than the nominal output power level.

NOTE

More details regarding the PTT limits can be found in section "PTT limits for various GSM bands" on page 106.

NOTE

The nominal output power levels and the corresponding power control levels are provided in section "Power control levels and tolerances" on page 103.

Entering and reading the lower range

NOTE

The entry fields shown below are for normal bursts. The same principles apply for random-access bursts.

NOTE

The time reference (t = 0) is the beginning of the first bit of the 148 bits of the normal burst. For details on a normal GSM burst, please check with section "Time slots and bursts" on page 98.

Again, the lower range is made up by a number of value pairs. Those value pairs give a position in time relative to the beginning of the burst and a power level. The blue graph of the PTT is achieved by plotting these value pairs and connecting them with lines.

Example: In the screen shot above, the following two value pairs are given:
4: 270 µs, −1 dB
5: 543 µs, −1 dB
This means: from 270 to 543 µs, the power transmitted by the mobile must not fall below its nominal output power level minus 1 dB.
The same limit for the power level applies from 20 µs after the beginning of the burst (value given in line 3).
From 543 µs (i.e. in the ramping-down phase of a burst) the minimum power level is not really an important criterion.

NOTE

More details regarding the PTT limits can be found in section "PTT limits for various GSM bands" on page 106.

NOTE

The nominal output power levels and the corresponding power control levels are provided in section "Power control levels and tolerances" on page 103.

Measurement limits

On all tests, the 4400 either displays a PASS or FAIL indication.
The bar display, available for many tests, gives a graphical orientation of the measurement values in relation to the test limits.
Those test limits are entered in this menu and the "Power measurement limits" menu.

How to read and enter the measurement limits

For every measurement parameter, there are lower and upper limits.
The **lower limit** gives the absolute minimum for a measurement result in order to pass the test. Any measurement result obtained being smaller than the lower limit will leave to a **FAIL** indication for that test.
The **upper limit** gives the absolute maximum for a measurement result in order to pass the test. Any measurement result obtained exceeding the upper limit will leave to a **FAIL** indication for that test.

For many parameters, the limits are symmetrical around the nominal value. In these cases, you can enter only one value which is applied to the negative and positive deviation from the nominal value. These limits are preceded by the "±" sign.

If a measurement can have positive result values only, the lower limit is 0 in most cases, especially for error rates where the optimum is 0 (no) error. For these measurements, only an upper limit (maximum) can be entered.

Example: The limits for the frequency error in the 900 MHz band have been set to ±90.0 Hz. This means that all measurement results obtained between –90.0 Hz and +90.0 Hz are within the range and will lead to a **PASS** indication for the test. If e.g. –90.1 Hz is obtained as a measurement result, the bar display will indicate a violation of the lower test limit and the entire test will be marked as **FAIL**.

Peak Phase Err	The peak phase error is the maximum phase error occurred within a single burst. This value can be both negative or positive. As the GMSK coding of signals used in GSM bases on a phase-shift of ±90° on bit transitions, the peak phase error limit should be considerably lower than that value.
Phase RMS	The RMS phase error is calculated from the phase errors of all bits within a burst, using the root mean square function. This value is always positive.

Uplink Time	Timing of the mobile in relation to the 4400. Downlink and uplink are shifted by three time slots in GSM, and there is also the timing advance the mobile is commanded to use. Therefore, the mobile's burst should arrive at the 4400 with a time delay of three time slots minus timing advance. The tolerance limit for this timing delay is entered here in μs. **Note:** A bit period in GSM is 48/13 μs (roughly 3.7 μs).
Frequency Err 900	The frequency error is computed as the slope of the regression line of the phase error over time. It tells you how well a mobile can synchronize to a base station. This entry field allows you to specify the symmetrical limit for GSM 900 systems. GSM 900 specifications specify a maximum permissible frequency error of about ±90 Hz. **Note:** This field is also applicable to GSM 850 systems.
Frequency Err 1800	The frequency error is computed as the slope of the regression line of the phase error over time. It tells you how well a mobile can synchronize to a base station. his entry field allows you to specify the symmetrical limit for GSM 1800/1900 systems. GSM 1800 specifications specify a maximum permissible frequency error of about ±180 Hz. **Note:** This field is applicable to GSM 1800 and GSM 1900 systems.
Burst Length	The length of a burst is critical in GSM. If a burst is too short, not all data transmitted by the mobile can be received. If it is too long, interference will be caused in the subsequent time slot. A standard GSM burst contains 148 bits, with a bit period of 48/13 μs. Consequently, a normal burst is some 546.5 μs long and has to fit into a time slot with a duration of 577 μs.
BER Ia	The **B**it **E**rror **R**atio is defined as the ratio of falsely decoded bits in relation to all bits transmitted. It is calculated for the three different bit classes separately. In this entry field, you enter the upper limit for the BER on class Ia bits.
BER Ib	The **B**it **E**rror **R**atio is defined as the ratio of falsely decoded bits in relation to all bits transmitted. It is calculated for the three different bit classes separately. The maximum value for the BER on class Ib bits is entered in this entry field.
BER II	The **B**it **E**rror **R**atio is defined as the ratio of falsely decoded bits in relation to all bits transmitted. It is calculated for the three different bit classes separately. Here you enter the upper limit for the BER on the class II bits.

RBER Ib	The **R**esidual **B**it **E**rror **R**atio is similar to the BER with the following exception: all frames the mobile detects an error in are rejected and not used for the RBER. Thus, only those frames are included in the RBER that were accepted by the mobile. The RBER is calculated for bit classes Ib and II only. Here you enter the permissible maximum for the RBER on the class Ib bits.
RBER II	The **R**esidual **B**it **E**rror **R**atio is similar to the BER with the following exception: all frames the mobile detects an error in are rejected and not used for the RBER. Thus, only those frames are included in the RBER that were accepted by the mobile. The RBER is calculated for bit classes Ib and II only. Here you enter the upper limit for the RBER on the class II bits.
FER	The **F**rame **E**rasure **R**atio is defined as the ratio of the frames rejected by the mobile in relation to all frames transmitted. A frame is marked as 'erased' and rejected by the mobile when its error detector finds an error. To do so, the mobile will use the checksum provided with the class Ia bits.
FBER	The Fast BER is similar to a BER measurement on the class II bits. The main difference is, that in Fast BER, all bits are unprotected. To measure the FBER, a special loop function is used (the so-called type C loop).

NOTE

A separate screen is available for **power level** limits.

Softkey for measurement limits

The following softkey eases limit handling:

Default Limits	Push this softkey if you want to return to the default limit values. The present values will be replaced by standard values.

Power measurement limits

In GSM, there are different power steps that determine the nominal RF output power for a mobile. While a mobile close to a base station will work on a lower power level, a mobile on the boundary of a cell will need a higher RF power level in order to make the base station 'hear' it.

The tolerances for the deviation of the peak power level from the nominal output power varies; a list of ETSI specification values can be found in section "Power control levels and tolerances" on page 103.

The effect of the power limits is as follows: On the traffic channel, the RF power of the mobile phone under test is measured and displayed on the GSM basic screen, the phase/frequency screen and the burst screen. The measured power

should be the nominal power level (MS Pwr setting), with a tolerance set on the Power Limits screen. If the measured power exceeds the limit, a FAIL is determined and displayed.

Example: If the MS Pwr setting is 23 dBm and the limit is 3 dB, then a measured value of 20.1 dBm is within the limits and the result is PASS. If, however, the measured value is 19.8 dBm, then the result exceeds the limits and the 4400 displays FAIL.

The Power Limit screen is divided into two sections for the lower (800 to 1000 MHz) and the upper (1800 to 2000 MHz) frequency bands.

For each power control step in a frequency band, the nominal output power level is given. An entry field allows you to specify the limit for this power level step. The default values reflect the limits for GSM 900 or GSM 1800 or GSM 1900, respectively.

Push this softkey if you want to return to the default limit values. The present values will be replaced by standard values.

NOTE

The GSM 900 limits apply to P-GSM 900, E-GSM 900, R-GSM 900 and also to GSM 850.

NOTE

Separate limits and defaults exist for GSM 1800 and GSM 1900. If the selected upper band is GSM 1800, the limits for this band apppear; if the upper band is GSM 1900, the respective limits for this band are shown. The upper frequency band can be changed on the "Basic system parameters" screen.

Coupling loss   When inserting cables, splitters, antennas or other RF equipment between the mobile under test and the 4400, there will always be some attenuation or coupling loss. In case of e.g. the universal antenna coupler, that attenuation may vary with both the type of mobile being tested and the frequency.
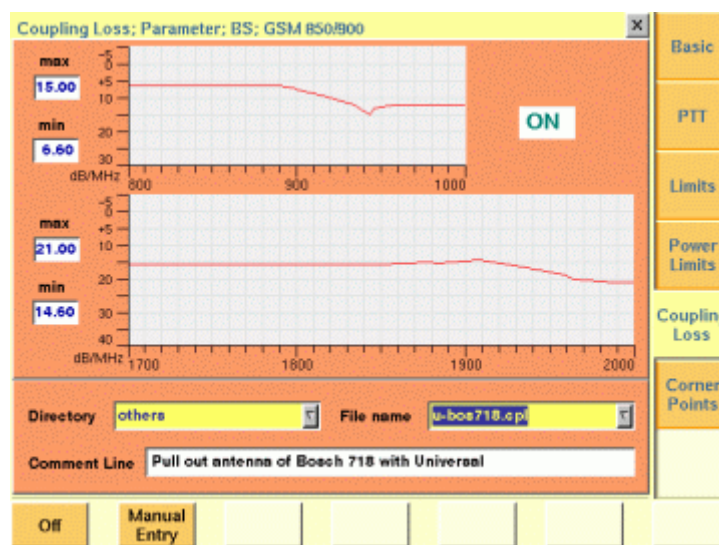
This menu allows to compensate that attenuation. To do so, a *.cpl file is required, containing the information about losses on specific frequencies. Those *.cpl files can be created either with the file editor within the RAPID! environment, on an external PC (and loading the file to the 4400 using a floppy disk), or from the Manual Input menu.

On the 4400, the contents of the currently selected *.cpl file will be displayed graphically for two frequency areas: The smaller, upper graph shows the coupling loss for the frequency range 800 to 1000 MHz, while the larger, lower graph shows the coupling loss for the frequency range 1700 to 2000 MHz.

On the left-hand side of either coupling loss graph, the 4400 provides two display fields informing you about the minimum and maximum attenuation for each frequency range.

The coupling loss parameters are stored in files. The file and directory structure allows you to store a set of parameters for each mobile and to locate the files for the phones from one vendor in one directory. The 4400 comes with a number of preinstalled directories for some of the larger manufacturers and a directory with parameter files for various "others". You can add more directories using the RAPID! file manager; the location of the coupling loss files and directories is /rapid/cpl.

**How to activate a previously
stored coupling loss definition**

File name	Comment Line
example.cpl	Coupling Loss: f(MHz), ampl.(dB)

Directory

Select the directory for the brand of the mobile.
This field allows you to browse the list of coupling loss
directories that are installed on the 4400. If you do not find
the manufacturer of the phone under test in the list of
directories, check the files in the "others" directory. The
4400 will allow you to browse through the files in the
selected directory using the File name scroll field.

File name

Select the corresponding file name on this field.
The 4400 will load the *.cpl file and display the data stored
graphically. Any comments or special instructions saved in
the *.cpl file will be displayed on the Comment Line field.
The files in the list are in alphabetical order; the uppercase
letters appear before the lowercase letters.
Coupling loss files created by the Manual Input feature (see
below) have a dot at the beginning so that they can be eas-
ily found at the beginning of the file list.

ON

Push this softkey to activate the coupling loss compensa-
tion. The display field on the right top of this menu will
change to ON.
The headline in all test masks will now carry a clear indica-
tion that the coupling loss compensation has been acti-
vated: Coupling Loss will be shown in red letters.

**How to store a coupling loss
definition on the 4400**

1 Save the *.cpl file on a floppy disk.

> **NOTE**
>
> The coupling loss compensation parameters for leading mobile phone types
> can be downloaded from the web (see above).

2 Insert the floppy disk into the 4400's disk drive.

3 Call the Welcome menu on-screen, using the (Esc) key.

4 Start RAPID! by first pushing the key and then the [RAPID!] marker
   tab.
   This will call the file manager on-screen.

5 Using the file manager, navigate to the /io/floppy directory.

6 Select the file to be stored on the 4400.

7 Push the **ETC** softkey and then the **Copy** softkey. This will call the File Copy
   to menu on-screen.

8 Enter the path and the file name to store the file, e.g. **/rapid/cpl/
   nokia/filename.cpl**.

NOTE

The maximum length for `file name` is 8 characters.

9 To confirm the entry and to finally copy the file, push the **OK** softkey.

10 A push on the **Exit** softkey will leave RAPID! and take you back to the Welcome menu of the 4400.

How to edit a coupling loss definition already stored on the 4400

1 Call the Welcome menu on-screen using the (Esc) key.

2 Start RAPID! by first pushing the ⚒ key and then the [RAPID!] marker tab.
   This will call the file manager on-screen.

3 Using the file manager, navigate to the `/rapid/cpl` directory.

4 Select the file to be edited.

5 Push the |Open| softkey and then the [Edit] marker tab to start the editing process.

The following issues should be considered when using coupling loss files:

– The *.cpl files are organized as tables.

– An optional comment may be placed in the first line. A comment is indicated by two slashes: `//comment`

– There are two sections within the *.cpl file, one for the frequency range 800 to 1000 MHz and one for the frequency range from 1700 to 2000 MHz.

– Every section must contain at least one line. The maximum for each section is 10 lines.

– Every line consists of two values: the frequency in MHz and the corresponding coupling loss in dB. Those two values are separated by a comma (,).
  Example:
  ```
  //Motorola P7389
  825.0,15.0
  1750.0,19.0
  ```

– Push the [File] marker tab to return to the file manager.

– Push the **Close** softkey to terminate any operation on the file previously opened.

– After the editing has been completed, push the **Save** softkey to store the file.

– A push on the **Exit** softkey will leave RAPID! and take you back to the Welcome menu of the 4400.

Entering coupling loss values from the Manual Input menu

This procedure allows you to enter coupling loss values and frequencies in a menu rather than invoking a text editor:

– On the Coupling Loss menu, select a coupling loss file that you want to change. Alternatively, select one on which you want to base your own file.

- Push the **Manual Entry** softkey within the Coupling Loss menu.
As a result, the Manual Input menu appears. This menu allows you to enter or change pairs of frequencies (in MHz) and coupling loss values (in dB) for downlink and uplink.

- Change values by overwriting them. Ensure to have at least one set of values for each of the two bands (800 to 1000, 1700 to 2000 MHz).

User defined Attenuation;Parameter;BS;GSM 850/900/1800						×	Basic
GSM 900			**GSM 1800/1900**				PTT
Frequency	Attenuation		Frequency	Attenuation			
800.0 MHz	1.5 dB		1700.0 MHz	2.0 dB			Limits
840.0 MHz	1.5 dB		1720.0 MHz	2.0 dB			
860.0 MHz	1.5 dB		1780.0 MHz	2.0 dB			Power Limits
870.0 MHz	1.5 dB		1800.0 MHz	2.0 dB			
890.0 MHz	1.5 dB		1820.0 MHz	2.0 dB			
940.0 MHz	1.5 dB		1890.0 MHz	2.0 dB			Coupling Loss
960.0 MHz	1.5 dB		1978.0 MHz	2.0 dB			
978.0 MHz	1.5 dB		1990.0 MHz	2.0 dB			Corner Points
990.0 MHz	1.5 dB		1995.0 MHz	2.0 dB			
995.0 MHz	1.5 dB		2000.0 MHz	2.0 dB			
Save	Save as..		Delete Values				

Save

When pushing the **Save** softkey, the entered values will be stored in the previously selected file.

Save as..
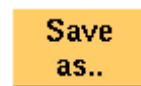
You can also give the file a different name by pushing this softkey. The 4400 then prompts you for the file name. The file name extension is added automatically. The file name also gets an additional "." at the beginning so that the manually created files appear at the beginning of the file name list on the Coupling Loss screen. This list is ordered by alphabet.
Note: All coupling loss files are stored in the `/rapid/cpl` directory.

Delete Values

You can delete a row of entered values by placing the cursor in one of the three fields of that row and then pushing this softkey.

After pushing the **ESCAPE** key, the Coupling Loss screen returns. You can now select the saved file from the list of coupling loss files.

NOTE

If you manually enter values and save them, a previously existing file named `.manual.cpl` will be overwritten.

NOTE

GSM 850 channels (channels 128 through 251) can be entered in the GSM 900 section.

Corner points On this menu, you may set eight points for checking critical burst positions. Those positions are defined by a time relative to the start of the burst, a minimum and a maximum RF power level.
The eight critical burst positions are shown graphically in the lower area of this menu. If burst measurements have been obtained, those positions will be shown in relation to the burst as in the example below.
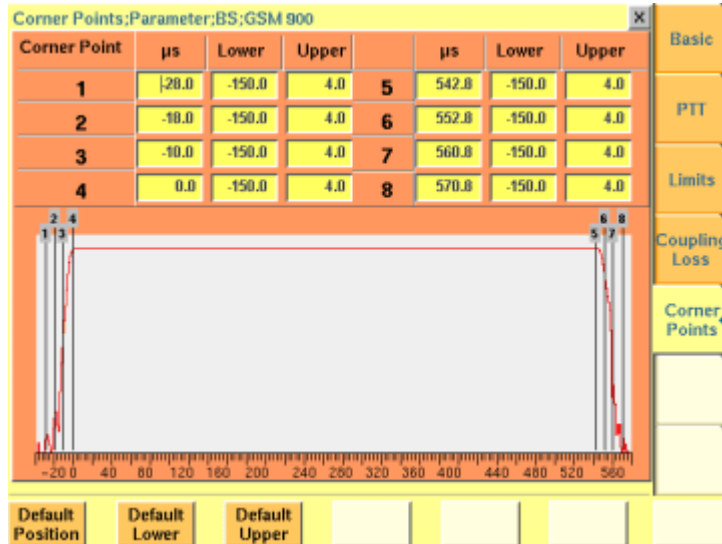
The current measurement results for those eight corner points are displayed in the "Burst test menu".
All measurement results obtained for a burst test have to be within the limits as entered on this menu in order to make the entire burst test PASS.
If a single corner point limit in only one of all bursts measured is violated, the whole burst test will be marked as FAIL.

NOTE

This menu bears a great saving in test time when used together with the remote control of the 4400. Using the corresponding SCPI commands, you may set the corner points remotely and read out the test results for those eight critical positions with one single command.

The meaning of the entry fields

Corner Point	µs	Lower	Upper		µs	Lower	Upper
1	-28.0	-150.0	4.0	5	542.8	-150.0	4.0
2	-18.0	-150.0	4.0	6	552.8	-150.0	4.0
3	-10.0	-150.0	4.0	7	560.8	-150.0	4.0
4	0.0	-150.0	4.0	8	570.8	-150.0	4.0

Corner Point 1...8 Each of those eight lines defines the position and the limits for one of the up to eight corner points.
Note: In most cases, the corner points are used to check critical positions within the ramping phase of an uplink burst.

μs	The values entered on these fields define the position of the corner points in μs relative to the beginning of a burst. The beginning of the burst is defined as the beginning of the first of the 148 data bits of a burst (see section for reference).
Lower	Here you enter the minimum RF power level the mobile has to transmit on. Any power level **equal to or above** the level entered here will be in the compliant range while any level below will violate the test limits and cause the entire test to be marked FAIL.
Upper	These are the entry fields for the maximum RF power level the mobile is allowed to transmit on. Any power level **equal to or below** the level entered here will be in the compliant range while any level exceeding the limit will violate the test limits and cause the entire test to be marked FAIL.

Softkeys of the corner points menu

Default Position — A push on this softkey will reset the positions of all eight corner points to the default values.
The eight default values are:
```
-28.0 µs
-18.0 µs
-10.0 µs
0.0 µs
542.8 µs
552.8 µs
560.8 µs
570.8 µs
```

Default Lower — To reset all values for the lower RF power limit to the default value, simply push this softkey.
The default value for all entry fields is `-150.0 dB.`

Default Upper — Push this softkey for resetting all upper RF power level limits to the default value.
The default value for all entry fields is `4.0 dB.`

GSM basics

In this section, you will find a short introduction to GSM systems and several overviews on important system parameters.

We will look at the following aspects of a GSM system:

– "Anatomy of a GSM network" – The structure and organisation of a GSM network.
– "Basic specifications of GSM bands" – a short table summarising the key specifications of the various GSM bands.
– "Time slots and bursts" – Here you will find more about the key GSM element: the time slot.
– "Signaling" – We've tried to translate a signaling table into 'normal' language to make it more easy to understand...
– "Technical requirements and parameters tested" – this page will introduce you to the reasons for testing specific functions and meaning of certain parameters.
– "Coding of MCC and MNC" – This section provides you with a number of examples regarding the coding of the MNC and MCC.
– "Power control levels and tolerances" – These tables give all the information on the power control levels and the related tolerances for the various GSM bands.
– "PTT limits for various GSM bands" – Here you will find the PTT and the related limits for the various GSM bands.
– "Voice coding" – A short introduction to GSM voice coding, bit classes and diagonal interleaving.
– "MS power classes" – This table gives all the information on the coding of the MS power classes for various GSM bands.
– "Coding of RX Qual" – Here you will find a table explaining the coding of the parameter RX qual, transmitted by the mobile with a measurement report.
– "Coding of RX Lev" – Here you will find a table explaining the coding of the parameter RX lev, transmitted by the mobile with a measurement report.

These sections only provide you with a very basic overview. If you would like to know more about GSM, we kindly recommend to check the following book:

Introduction to GSM by Siegmund Redl, Matthias Weber and Malcolm W. Oliphant
A comprehensive, detailed and yet easily understandable introduction to theory and practice of GSM technology.
Publisher: Artech House, London, UK. ISBN 0-89006-785-6.

Anatomy of a GSM network

GSM is a so-called **cellular network**. This means that the geographical area covered by the network is subdivided into cells. The geographical size of such a cell is determined by a radius of between some 100 meters and up to 30 km, depending on the expected use within the cell.

As long as a **mobile** is within the geographical area of a cell, it can call the network and it can be paged by the network.

A quite complex hand-over mechanism ensures that the call will not be terminated whenever a mobile leaves the geographical area of one cell and moves into another one. In GSM, this usually works even when you are crossing borders!

Each cell is managed by a **base station (BS)**. This is the relay between the telecommunications network and all the mobiles within its geographical area.
The physical link between BS and mobile is provided by a radio interface (the so-called **Um interface**), while the interface between the BS and the network usually is an ISDN-link (the so-called **A–bis interface**).
The BS is taken care of by the **base station controller (BSC)**.

The BSC mainly coordinates the radio traffic within a cell by e.g. assigning traffic channels. There might be base stations with their own dedicated BSC already integrated and there might be BSCs taking care of several base stations.
The BSCs again are managed by the "big boss" within a network: the **mobile station switching center (MSC)**. The MSC administers all calls within its region, it knows which mobiles are located there, checks their authentication and will pass on information "to the top" in order to make it possible for the telecommunications network to page a mobile and to put through a call. The MSC is therefore directly linked to the telecommunications network.

With every single call a mobile makes or receives, all the instances (BS, BSC, MSC) are involved.

Going for a **call**, checking **authorization**, assigning **traffic channels**, changing **cells** – this all requires an intensive exchange of information between mobile, BS, BSC and MSC. This exchange of information, required to set up or put through a call, is what we refer to as "Signaling".

Basic specifications of GSM bands

The table below gives the specifications for the various GSM bands (based on GSM specification 05.05).

	GSM 450	GSM 480	GSM 850	E–GSM 900	R–GSM 900	DCS 1800	PCS 1900
Frequency range uplink (MS –> BS)	450.6 to 457.4 MHz	479.0 to 485.8 MHz	824.2 to 848.8 MHz	880.2 to 914.8 MHz	876.2 to 914.8 MHz	1710.2 to 1784.8 MHz	1850.2 to 1909.8 MHz
Frequency range downlink (BS –> MS)	460.6 to 467.4 MHz	489.0 to 495.8 MHz	869.2 to 893.8 MHz	925.2 to 959.8 MHz	891.2 to 959.8 MHz	1805.2 to 1879.8 MHz	1930.2 to 1989.8 MHz

Channel numbers	259 to 293	306 to 340	128 to 251	0 to 124 and 975 to 1023	0 to 124 and 955 to 1023	512 to 885	512 to 810
Duplex spacing	10 MHz	10 MHz	45 MHz	45 MHz	45 MHz	95 MHz	80 MHz

Channel spacing	200 kHz
Channel utilization	TDMA (**T**ime **D**ivision **M**ultiple **A**ccess. Every channel is split into eight time slots. This allows a maximum of eight calls simultaneously on the same frequency channel.
Length of a time slot	$(156¼*48)/13$ µs = 577 µs. Eight time slots form a frame (= 4.615 ms).
Modulation	GMSK (**G**aussian **M**inimum **S**hift **K**eying). Phase shifts of ±90° on bit transitions.

Time slots and bursts GSM is a TDMA system (TDMA: **T**ime **D**ivision **M**ultiple **A**ccess). This means that the same physical channel (i.e. frequency) is used for several calls at the same time. To avoid data collision, all transmissions have to be in a precise timely synchronization. Therefore, a mobile or BS is only allowed to transmit data during its **time slot**. Outside its time slot, the mobile must not emit 'any' power. Consequently, a mobile has to ramp up its power very quickly (some 30 µs) from zero to nominal. And once the data has been transmitted, it must abruptly decrease power again. This radio pulse, transmitted within a time slot, is what we call a **burst**.

In GSM, the length of a time slot is 156¼ bit periods or 577 µs. Eight time slots (numbered from 0 to 7) form a TDMA frame (many times just simply called a 'frame') that is then 4.615 ms long.
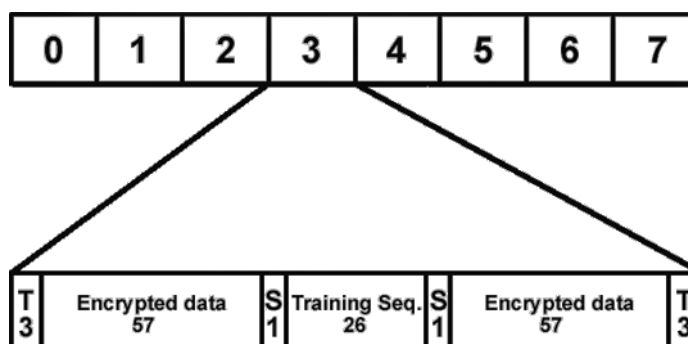
During a time slot, 148 'useful' bits are transmitted. 114 bits out of that are 'real' data. The remaining bits are used to avoid timing conflicts and to synchronize the receiver to the bit stream. The meaning of the different data transmitted within a time slot is given below.

T (tail bits)   The three tail bits bear no information. Consequently, no data will be lost if the burst does not exactly "catch" the time slot or if the bursts of the mobiles transmitting in the previous or subsequent time slots are slightly too long or not perfectly synchronized.

Notes:

– The propagation time required by a radio wave to travel 1 km is approx. 3.3 µs (the duration of one bit in GSM is approx. 3.7 µs).

– Usually, mobiles are on variable distances to the base station.

– To avoid that a burst of a mobile 'comes in too late', the mobile is commanded by the base station to use a timing advance. This means that the mobile will send its burst some microseconds earlier. After the propagation delay, it will then arrive at the base station in due time.

– The 'timing advance' is explained in detail below.

S (stealing flag)   The stealing flag will be set when a burst does not contain call data but signaling data.

Training Seq.   The training sequence is a bit sequence known by both transmitter and receiver. The receiver of a burst looks for this sequence in the data stream. Thus, the receiver can easily allocate the useful data within the burst.

Encrypted data   This is the 'payload' of the burst, often also called the useful data. The data transmitted here is

– encrypted to safeguard it against interception,

– partly encoded so that incorrectly transmitted bits can be detected and even corrected.

Note: Please refer to section "Voice coding" for more details.

Timing advance   As said above, timing is extremely critical in GSM. All transmissions between mobile and base station have to be perfectly synchronized. Due to the RF signal propagation time (some 3.3 µs per km distance), the mobile receives the signals transmitted by the base station with a short time delay. Of course, the mobile will synchronize to the base station's timing. However, it has no means of detecting the delay due to signal propagation, so it will be back in timing. Should the mobile transmit now, the signals will arrive at the base station 'too late' and may cause interference in subsequent time slots. To avoid that, the base station measures the delay of the bursts received from the mobile in relation to its internal timing and then commands the mobile to send its bursts 'a little' earlier. In GSM terms, this is called 'assigning the mobile a timing advance'.

Example: Let's assume that there has been no timing advance set. The mobile is 10 km away from the base station. Due to signal propagation time, the bursts transmitted by the base station will arrive at the mobile with a delay of some 33 µs. The mobile will synchronize to that and will therefore be 33 µs back in timing without realising it. Should the mobile transmit a burst, that burst would also travel 33 µs. As a consequence, the burst would arrive at the base station 66 µs or some 18 bit periods too late. This would lead to severe interference in the subsequent time slot.

To avoid that interference, the base station will command the mobile to send its bursts 66 µs or 18 bit periods earlier to compensate the RF signal propagation delay.

NOTE

As the mobile has not been assigned a timing advance when it transmits random-access bursts, these bursts have to be significantly shorter than normal bursts to avoid interference in the subsequent time slot.

NOTE

A bit period in GSM is 48/13 µs long (some 3.692 µs).

NOTE

Rule of thumb: A distance of one km corresponds to a signal propagation time of approx. one bit period and a required timing advance of approx. two bit periods.

Signaling

Signaling could be thought of as the red tape in GSM radio communication. So it is no wonder if the term strikes you as being somewhat mysterious and obscure. Here is an example to shed some light on the subject.

MS call

A MS wants to set up a call in the example below. The signaling necessary to set up that call is 'translated' into 'common language' to make the procedure more understandable. During a MS call, the basic signaling between your 4400 and the unit under test would be quite similar.

MS	BS	Message	Meaning
→		Channel_Request	I want something from you.
←		Immediate_Assignment	Speak out! On channel 44, timeslot 3.
→		Service_Request	I want to make a call.
←		Authentication_Command	The MSC says you must prove your identity. What's the result of authentication procedure x?
→		Authentication_Completed	I'm OK. The answer is 42.
←		Ciphering_Command	Correct. Let's continue speaking encrypted. The parameter for encryption is 12345.
→		Ciphering_Completed	✿■✳❙ ◇❏❏✳⊕❏✳❙✎ Encryption is on.
→		Setup	Dialled number xy. This is what I can do: I can fax, but sorry no half rate.
←		Call_Proceeding	Understood. MSC and the others are trying to get in touch with the person at the other end.
←		Assign_Command	Your TCH is channel 16, full rate.
→		Assign_Completed	Thanks a lot. I told my RF parts.
←		Alert	It's ringing. Wait till he picks up the receiver.
←		Connect	The call is through. Now he's there.
→		Connection_Acknowledged	Thanks. See you at the next measurement report.

NOTE

Numbers and symbols above are just used as illustrative examples.

Technical requirements and parameters tested

Testing in a GSM radio network is quite different compared to testing in analog networks. Some of the reasons for that are:

- GSM is a TDMA system. As for all TDMA systems, the perfect synchronization in time is essential. Therefore, the timing and form of the burst are important test criteria.

- GSM uses the GMSK modulation to transmit data. This modulation only works properly when phase and frequency error do not exceed certain limits.

- The transmitters only work in burst mode. Consequently, the peak power has to be measured.

- All information transmitted is digital. So the signal/noise ratio is no longer a criterion for judging a receiver, but the level of falsely detected bits is.

- Call setups as well as many more functions are software-driven and based on signaling tables. Any test of a radio connection therefore incorporates also a test of the signaling software of the unit under test.

What the CCITT is for analog radio communication, the GSM Specifications are for GSM. The GSM Specifications define the maximum deviations from the expected values allowed for a mobile under test and how to measure those deviations. Please find below some of the most important GSM test parameters, their meaning and the usual test limits.

Phase error	The phase error in rgeneral provides a quality reading of the mobile's modulator.
	– The **peak** phase error is the maximum phase error occurred within a single burst.
	– The RMS phase eror is calculated from the phase errors of all bits within a burst, using the root mean square function.
	– In GSM, the maximum phase error must not exceed 20° and the RMS phase error should be less than 5°.
	Note: GMSK codes bit transitions by phase alterations of $\pm 90°$.
Frequency error	The frequency error is computed as the slope of the regression line of the phase error over time. It tells you how well a mobile can synchronize to a BS.
	The GSM 900 specifications define a maximum permissible frequency error of ± 90 Hz, while GSM 1800/GSM 1900 specifications allow ± 180 Hz.
Peak power	In GSM, the transmitted peak power is divided into power classes. A mobile close to the base station with will have to transmit at a lower power level than a mobile at the boundary of a cell.
	The power classes and the corresponding tolerance limits for various GSM bands can be found in section "Power control levels and tolerances" on page 103.
power/time template	In order not to create interference in other time slots and to ensure that there is a constant power level during the burst period, a burst sent by any mobile has to fit precisely into a so-called power/time template.
BER (Bit Error Ratio)	The BER is the ratio of falsely decoded bits in relation to all bits transmitted.
	GSM uses different bit classes. The main difference between those bit classes is the level of error-protective coding during transport over the radio interface.
	Class Ia bits are very significant bits that are well protected.
	Class Ib bits are still significant bits, but not so well protected.
	Class II bits are not protected at all. Consequently, these bits are often used for a quick check of the BER.
	The Fast BER is similar to a BER, but using a special loop with uncoded bits (the so-called type C loop).
FER (Frame Erasure Ratio)	The FER is defined as the ratio of the frames rejected by the mobile in relation to all frames transmitted.
	A frame is marked as 'erased' and rejected by the mobile when its error detector finds an error. To do so, the mobile will use the checksum provided with the class Ia bits.

RBER (Residual Bit Error Ratio) The Residual Bit Error Rate is quite similar to the BER (see above). The only difference: erased frames (see above) are not taken into account for calculation. The RBER is measured for bit classes Ib and II only.

Coding of MCC and MNC

The MCC (= **M**obile **C**ountry **C**ode) is used to give the country in which the mobile subscriber is registered in. In order to do so, every country is assigned a unique three-digit number. Those codes are given in the table below.

Within those countries, there might be several national networks. Those national networks are coded using a two-digit number (three digits in North America), the so-called MNC (= **M**obile **N**etwork **C**ode).

> **NOTE**
>
> The list below contains just a few examples for your information.

Country	MCC	MNC	Operator name	Handset code	Band
Albania	276	01	Albanian Mobile Communications	AMC-AL	900
Albania	276	02	Vodafone Albania	voda-AL	900/1800
Australia	505	01	Telstra MobileNet	Telstra	900/1800
Australia	505	02	Singtel Optus Ltd. (YES OPTUS)	Optus	900
Australia	505	03	Vodafone Pacific Pty Ltd.	VFONE AUS	900/1800
Canada	302	370	Microcell Telecom Inc	MCELL	1900
Canada	302	720	Rogers Wireless		1900
Denmark	238	01	TDC Mobil A/S	DK TDC	900/1800
Denmark	238	02	Sonofon	SONO	900/1800
Denmark	238	20	Telia A/S Danmark	Telia	1800
Denmark	238	30	Orange A/S	Orange	900/1800
Vietnam	452	01	Vietnam Telecoms Services Co GPC (Mobifone))	452 01	900
Vietnam	452	02	Vietnam Telecoms Services Co (Vinaphone)	452 02	900

Power control levels and tolerances

GSM 450, GSM 480, GSM 850 and GSM 900 systems

Power control level	Nominal output power (dBm)	Tolerance (dB)
0	43 (GSM phase 1 only)	±2

1	41 (GSM phase 1 only)	±2
0...2	39	±2
3	37	±3
4	35	±3
5	33	±3
6	31	±3
7	29	±3
8	27	±3
9	25	±3
10	23	±3
11	21	±3
12	19	±3
13	17	±3
14	15	±3
15	13	±3
16	11	±5
17	9	±5
18	7	±5
19...31	5	±5

GSM 1800 systems

Power control level	Nominal output power (dBm)	Tolerance (dB)
29	36	±2
30	34	±3
31	32	±3
0	30	±3
1	28	±3
2	26	±3
3	24	±3
4	22	±3
5	20	±3
6	18	±3
7	16	±3

8	14	±3
9	12	±4
10	10	±4
11	8	±4
12	6	±4
13	4	±4
14	2	±5
15...28	0	±5

GSM 1900 systems

Power control level	Nominal output power (dBm)	Tolerance (dB)
22...29	reserved	reserved
30	33	±2
31	32	±2
0	30	±3
1	28	±3
2	26	±3
3	24	±3
4	22	±3
5	20	±3
6	18	±3
7	16	±3
8	14	±3
9	12	±4
10	10	±4
11	8	±4
12	6	±4
13	4	±4
14	2	±5
15	0	±5
16...21	reserved	reserved

PTT limits for various GSM bands

Depending on the GSM band and the mobile's peak power level, GSM specifications define different limits for the PTT.

These limits can be found in the table below.
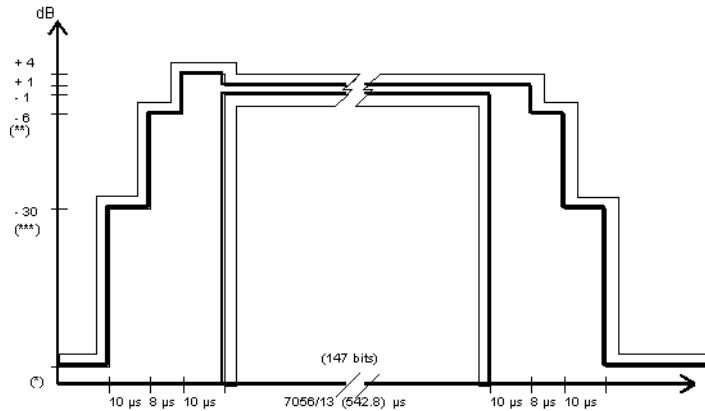
Comments for Normal bursts and Random-access bursts

	GSM 400, GSM 850 and GSM 900	GSM 1800	GSM 1900
(*)	Between the active bursts, the residual output power shall not exceed the following levels: –59 dBc or –54 dBm, whichever is the higher. **Exception:** For the time slot preceding the active slot, the limit is –59 dBc or –36 dBm, whichever is the higher.	Between the active bursts, the residual output power shall not exceed the following level: –48 dBc or –48 dBm, whichever is the higher.	Between the active bursts, the residual output power shall not exceed the following level: –48 dBc or –48 dBm, whichever is the higher.
(**)	The limit of –6 dBc shall **not** be used in case of: – power control level 16: limit is –4 dBc instead; – power control level 17: limit is –2 dBc instead; – power control levels 18 and 19: limit is –1 dBc instead.	The limit of –6 dBc shall **not** be used in case of: – power control level 11: limit is –4 dBc instead; – power control level 12: limit is –2 dBc instead; – power control levels 13, 14 and 15: limit is -1 dBc instead.	
(***)	The limit shall either be –30 dBc or –17 dBm, whichever is the higher.	The limit shall either be -30 dBc or –20 dBm, whichever is the higher.	

NOTE

Details regarding power control levels for all GSM bands can be found in section "Power control levels and tolerances" on page 103.

Voice coding   In GSM, the call data are digital. When you talk into the microphone of a mobile phone, your words are converted into bits. These bits could then be used to directly modulate the transmitter and thus be sent to the base station. However, things are not so simple in GSM.

Slight interference on the radio link could alter bits, make bits illegible or even block entire frames. As a result, the conversion of the bits transmitted into voice could be corrupted or even impossible. As interference tends to appear in random fashion, it might just affect the most significant bits. The following example will illustrate what can happen if bits disappear.

Example: Take the word 'tester' and remove individual letters. Try then to deduce what the original word was from the fragments that are left. You will find, for example, that the first 'e' carries much more significant information than the second one. Even if you delete 'er', you still realise that the original word has something to do with 'test', 'testing' or 'tester'. But if you take away the 'te', the remaining fragment of the original word could stand for all kinds of things (sister, mister, oyster, Easter...).

So what is done in GSM is to try at all costs to get those bits from transmitter to receiver that are significant for the understanding. Therefore, the bits are arranged in three classes:

– class Ia – very significant bits, must be protected

– class Ib – significant bits, should be protected

– class II – bits that may be distorted without affecting speech quality too much

In GSM fullrate, the voice codec will convert 20 ms of speech into 260 'raw' data bits. 50 of those will be class 1a, 132 are class Ib and the remaining 78 are class II.

First, the class Ia and Ib bits are converted into so-called block code. This procedure adds 3 check bits for the class Ia bits and 4 check bits for the class Ib bits. The check bits will be used by the receiver's decoder to check whether the block is error free. Consequently, block coding is a means of error detection.

The 182 bits of classes Ia and Ib are converted into 378 bits by a mathematical process (convolutional coding), adding redundancy. This is like formulating an important statement in several different ways, so that the information will be saved in at least one of its forms. During the decoding process, the intact parts of the different forms of the statement are looked for. So, convolutional coding is a means of error correction.

The 78 bits of class II are not protected at all. If they are erroneous, voice quality will suffer somewhat, but will still be adequate.
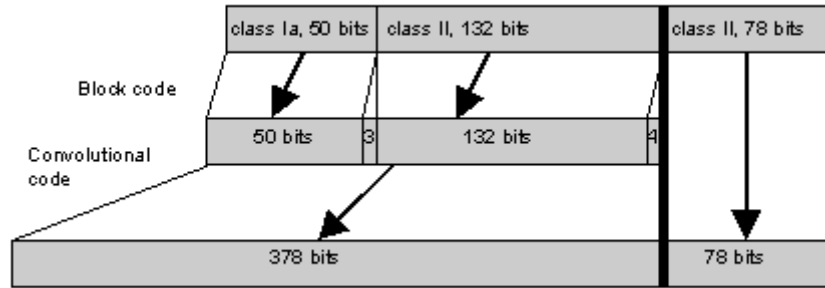
All the coding mentioned above expands the voice block from 260 to 456 bits. But the effects on quality and reliability of transmission are so positive that this is gladly accepted.

To transmit a voice block, it has to be spread out among several bursts. Here another means of error prevention is used, the so-called interleaving. Each voice block of 456 bits is split into eight subblocks of 57 bits each. Each subblock is transmitted with a single burst.
Using this complicated way of transmitting data, brief interference has little effect on a single voice block and the data can be retrieved by convolutional decoding.

Block diagonal interleaving
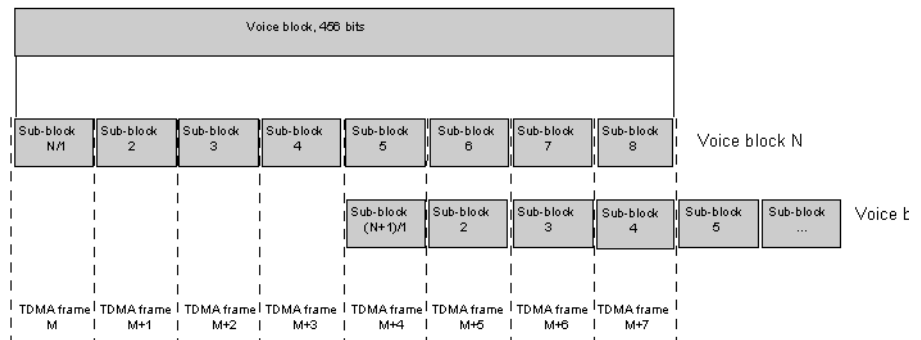
NOTE

When you perform RX tests, just compare the error ratios of bits of classes Ia and II when the connection is disturbed or at low signal levels. You will find that the error ratios for class Ia bits are whole orders less than for class II bits.

NOTE

A similar technique is used for coding the signaling bits; something called 'fire coding' in GSM terms. There, however, are no less significant bits. So all signaling bits are protected by convolutional coding.

MS power classes   The power classes of the mobiles vary between the GSM bands. The codes stand for different RF power levels in different GSM bands. The table below gives a short overview.

Power Class Code	1	2	3	4	5
GSM 900	43 dBm	39 dBm	37 dBm	33 dBm	29 dBm
PCN/GSM 1800	30 dBm	24 dBm	36 dBm		
PCS/GSM 1900	30 dBm	24 dBm	33 dBm		

Coding of RX Qual   Based on the number of bits corrected by its own channel decoder, the mobile estimates the BER. The parameter reported back by the mobile in its MS report is coded as follows:

RX Qual	BER in %
0	< 0.2
1	0.2...0.4
2	0.4...0.8
3	0.8...1.6
4	1.6...3.2
5	3.2...6.4
6	6.4...12.8
7	> 12.8

Coding of RX Lev   For the MS Info message, the mobile under test estimates the RF power level at which it receives the base station. The RX Lev parameter reported back by the mobile in its MS report is coded as follows:

RX Lev	RF power in dBm
0	<= −110
1	−109
...	...
k	−110 + k
...	...
62	−48
63	> −48

GPRS Tests

This section covers information on mobile phone testing in GPRS synchronous mode. Read "Essentials" to learn of GPRS test prerequisites and modes before starting a test using the 4463 GSM/GPRS System Option and 4468 EDGE System Option.

Essentials

In this section you will find essential information on GPRS test mode.

Why GPRS tests?

GPRS (General Packet Radio Service) is a new protocol, replacing the GSM protocol above the physical layer; GSM and GPRS operation share the same physical resources. In manufacturing and service, the purpose of testing is to ensure that the hardware is working correctly, rather than testing the protocol software.

Some GSM/GPRS mobile phones may have single-slot capability only. Although the RF hardware is the same for GSM and GPRS, and may already be tested in GSM mode, additional tests for GPRS may be advisable because the phone's driver software may use the hardware components differently in GPRS.

Most GSM/GPRS mobile phones support multiple time slots in the downlink and uplink. This capability challenges the mobile phone more than in single-slot operation as on a voice channel: Components may produce side effects due to voltage drops or additional heat when multiple slots are transmitted or received. Therefore, most manufacturers ensure product quality by multislot transmitter and receiver tests.

Uplink and downlink assignments

GPRS is used to transmit packets or blocks of data from one end to another. While circuit-switched connections are bidirectional and block a channel in both directions for the duration of a call, packet data transfers may be unidirectional and occupy the channel only for the time of the block transfer, i.e. the resource "channel" can be shared by many users and allocated to users on a block-by-block basis.

The channel and block administration is performed by the base station. When one party needs to transfer data, a so-called Temporary Block Flow (TBF) is initiated on RLC/MAC level. Since the information flow is unidirectional, there is a separate TBF for each direction, called Downlink TBF and Uplink TBF. Moreover, each TBF can use one or multiple time slots. A bidirectional data transfer can be asymmetric, i.e. may use more time slots in one than in the other direction.

The GPRS protocol uses a number of new protocol layers and procedures. For the purpose of RF testing, only RLC and MAC layers are required, plus a few procedures such as the GPRS Attach (registration procedure of a mobile in a GPRS network). Higher layers (e.g. LLC) and their procedures, such as PDP Context Activation, are not required for RF testing and therefore not supported by the 4463 GSM/GPRS System Option and 4468 EDGE System Option.

Assignment of time slots for uplink and downlink

A GPRS mobile phone is associated to a multislot class; each multislot class determines how many time slots the mobile can transmit and/or receive, and also defines the minimum duty cycle between transmit and receive slots. The most important parameters are:

– The maximum number of downlink slots (without any assigned channel in the uplink)

– The maximum number of uplink slots (without any assigned channel in the downlink)

– The maximum number of combined slots assigned in downlink and uplink at any time

Example: A mobile of multislot class 10 can use up to four time slots in the downlink and up to two time slots in the uplink, but only up to five time slots in total (e.g. 4 DL + 1 UL or 3 DL + 2 UL).

After a GPRS Attach, the multislot class is shown on the GPRS Basic menu together with these capabilities; in the above example, it would display "10: 4/2/5" meaning that the mobile is of multislot class, supporting 4 DL, 2 UL and 5 time slots in total.

NOTE

Changing the number of downlink or uplink slots is supported only when the mobile is GPRS attached. The Willtek 4400 only allows you to enter permissible combinations of downlink and uplink slots, depending on the multislot class.

Transmit Power

Uplink Power Control

In a real network, the transmit power of the mobile phone can be controlled by the network directly (close loop power control) and, depending on the setting of the open loop power control parameter Alpha (see below), partially also by the mobile phone (open loop power control). In case of the open loop power control, the transmit power depends on the received signal strength. The power levels are given as follows.

Power Levels

Generally, the power levels $Pn = Gamma_0 - Gamma_{CH}$ at the uplink slots, where n is the number of the burst within a TDMA frame are given by

$$Pn = min[Gamma_0 - Gamma_{CH} - Alpha * (C + 48), P_{max}]$$

where $Gamma_0$ = 39 dBm for GSM 900 and $Gamma_0$ = 36 dBm for GSM 1800/1900, $Gamma_{CH}$ is given in the table below, Alpha is a weight parameter of the open loop (Alpha = 0 without open loop, Alpha = 1 for open loop fully deployed), C is the power level of the base station received by the mobile, and P_{max} is the maximum output power of the mobile according to its power class. In the 4463 GSM/GPRS System Option and 4468 EDGE System Option, the parameter Alpha is set to 0, i.e. the mobile station power level is fully controlled by the 4463 GSM/GPRS System Option and 4468 EDGE System Option.

Gamma$_{CH}$ (Binary Coded)	Pn [dBm] GSM 900	Pn [dBm] GSM 1800/1900
0	39	36
1	37	34
2	35	32
3	33	30
4	31	28
5	29	26
6	27	24
7	25	22
8	23	20
9	21	18
10	19	16
11	17	14
12	15	12
13	13	10
14	11	8
15	9	6
16	7	4
17	5	2
18	3	0
19	1	−2
20	−1	−4
21	−3	−6
22	−5	−8
23	−7	−10
24	−9	−12
25	−11	−14
26	−13	−16
27	−15	−18
28	−17	−20
29	−19	−22
30	−21	−24

Gamma$_{CH}$ (Binary Coded)	Pn [dBm] GSM 900	Pn [dBm] GSM 1800/1900
31	−23	−26

For the purpose of transmitter tests, you will probably want to determine directly the transmit power level of the mobile phone. Therefore, open loop power control is disabled by the 4463 GSM/GPRS System Option and 4468 EDGE System Option (i.e. Alpha = 0). For measurements without open loop, the power levels Pn = Gamma$_{CH}$ − Gamma$_0$ translate into the values of the SCPI command `:CONF:GPRS:MSTation:GAMMa` (see Section 8, "SCPI").

A real network can even set different uplink power levels for the individual time slots assigned to the mobile. This is also supported by the 4463 GSM/GPRS System Option and 4468 EDGE System Option; the power levels for the individual uplink time slots can be defined on the GPRS Burst menu.

Downlink power

In the context of the GSM specifications, downlink power control means the capability of the base station to adjust its downlink power level from time slot to time slot; different modes and algorithms exist how to set the levels. The network informs the mobile phone of the used mode in order to help the automatic gain control software in the mobile phone. Of course, there are other factors that impact the received signal strength in the phone, such as variations over time due to reflections in the signal propagation path.

In order to test the receiver hardware of a mobile phone rather than its software, the 4463 GSM/GPRS System Option and 4468 EDGE System Option does not inform the mobile of the downlink power control parameters; it rather pretends that no downlink power control is being performed. Nevertheless, the 4463 GSM/GPRS System Option and 4468 EDGE System Option allows you to adjust the transmitted power level between the different time slots. The entry fields for individual slots can be found on the GPRS Receiver menus.

Test prerequisites

In order to be able to make tests in synchronous mode (i.e. with proper channel assignments), the mobile phone must be GPRS attached. The GPRS Attach is a procedure which usually informs the network of the mobile's presence and GPRS capabilities within a GPRS cell. The sequence of events during a GPRS Attach is as follows:

1 When the GSM/GPRS synchronous mode menus are active, the 4463 GSM/GPRS System Option and 4468 EDGE System Option simulates a GSM network with GPRS capabilities, i.e. it broadcasts in one of the System Information Messages (type 13) that it supports GPRS data transfer. The GPRS Attach field in the call graph of the GPRS Basic menu is shown with white background.

2 When the mobile is switched on, it scans the frequency bands for a valid network. When it finds and accepts the 4463 GSM/GPRS System Option and 4468 EDGE System Option GSM/GPRS network, it may (or may not) perform a Location Update for its GSM presence in the network.

3 The mobile phone will have to initiate a GPRS Attach procedure. Depending on the implementation in the mobile phone, this could be done in one of two ways:

– Some mobile phones initiate the GPRS Attach automatically after they selected a network which has GPRS capabilities.

– In other mobile phones, the GPRS Attach is initiated only when the user requests a GPRS service, e.g. by selecting an appropriate entry on the user interface of the mobile phone.

4 The 4463 GSM/GPRS System Option and 4468 EDGE System Option and the mobile will communicate via UL/DL TBFs to exchange protocol information. When the procedure is concluded successfully:

– The mobile phone will usually indicate that it is registered to a GPRS network (e.g. by the word "GPRS" in its display).

– The 4463 GSM/GPRS System Option and 4468 EDGE System Option will indicate the multislot class of the mobile in the Multislot class field on the GPRS Basic menu.

– The background of the GPRS Attach field in the call graph of the GPRS Basic menu will turn green.

– No TBF will be active.

NOTE

The GPRS Attach can be performed on both the GSM and the GPRS menus of the 4463 GSM/GPRS System Option and 4468 EDGE System Option GSM/GPRS System Option. The GPRS measurements, however, are supported from the GPRS menus only.

GPRS test modes The 4463 GSM/GPRS System Option and 4468 EDGE System Option GSM/GPRS System Option supports different combinations to test the transmitter and receiver of a mobile phone in GPRS mode.

TX

In the TX test mode, only the transmitter of the mobile phone is tested. The transmitter parameters such as power level, power versus time, frequency and phase error are measured for each time slot individually. The 4463 GSM/GPRS System Option and 4468 EDGE System Option shows the results for the selected time slot. The number of allocated time slots and the time slot to be measured are user-selectable.

For the TX test mode, the 4463 GSM/GPRS System Option and 4468 EDGE System Option forces the mobile station to request an Uplink TBF by way of the GPRS Test Mode Command (described in ETSI specification GSM 04.14).

BLER–BCS

In the BLER-BCS (BLock Error Rate based on the Block Check Sequence) test mode, only the receiver is tested; the mobile receives a lot of data blocks but transmits only occasionally (when polled).

For the BLER-BCS measurement, the 4463 GSM/GPRS System Option and 4468 EDGE System Option establishes a Downlink TBF and transmits blocks of data to the mobile phone. Every 50 blocks or so, the 4463 GSM/GPRS System Option and 4468 EDGE System Option requests, just like a real network, an indication from the mobile whether or not the data have been received correctly by the mobile phone. It acknowledges the proper reception or reports which blocks have been received in error. The phone can detect reception errors – after channel decoding – by comparing the transmitted Block Check Sequence (a checksum included with each block) with the calculated one. Finally, the BLER (Block Error Rate) is the number of reported errors compared to the whole number of received blocks.

The BLER is calculated over all time slots, in accordance with the ETSI specifications. Each measurement result is generated from a user-defined number of samples. A sample in case of the BLER-BCS measurement is a block, which in the worst case of one time slot assigned in the downlink, takes 20 ms. So if you select 200 samples, a measurement result will be returned every 4 seconds.

BLER–USF+TX

The advantage of the BLER-USF+TX test mode is that it supports receiver and transmitter measurements in parallel, which can improve the test time in critical environments such as a manufacturing line. The BLER-USF (BLock Error Rate based on the Uplink State Flag) receiver test is an ETSI-defined test method just like the BLER-BCS measurement, and its functionality can be compared with a paging test as follows.

The 4463 GSM/GPRS System Option and 4468 EDGE System Option first establishes an Uplink TBF (by way of the GPRS Test Mode Command). During the establishment procedure, the mobile phone is assigned an Uplink State Flag (USF), which is a temporary address used in the MAC header of each block in the downlink to assign an uplink block to an individual mobile. Every now and then, the 4463 GSM/GPRS System Option and 4468 EDGE System Option then sets the USF to that of the phone under test. When the mobile receives a block with the USF set to its own one, it has to transmit a block of data. If the mobile does not send that block of data, this is an indication that it did not receive the data block with the assigned USF from the 4463 GSM/GPRS System Option and 4468 EDGE System Option. So every missing block transfer from the mobile points to a block reception error. The number of missing blocks compared to the number of blocks transmitted by the 4463 GSM/GPRS System Option and 4468 EDGE System Option with a correct USF, indicates a block error.

A sample in case of the BLER-USF measurement is a block, which in the worst case of one time slot assigned in the uplink, takes 20 ms. So if you select 200 samples and one uplink time slot, a measurement result will be returned every 4 seconds.

GPRS tests and coding schemes   The 4400 uses CS-1 for the BLER-USF and BLER-BCS measurements. This is the coding scheme most commonly used by real networks.

Performing GPRS tests   In addition to GSM measurements, the 4463 GSM/GPRS System Option and 4468 EDGE System Option GSM/GPRS System Option allows multislot measurements in GPRS test modes introduced in "GPRS test modes" on page 114.

To access GPRS test mode, press the **Calls & Meas.** softkey on the Welcome menu and then press the **GPRS...** softkey on the GSM Basic menu. The GPRS Basic menu will appear.

The 4463 GSM/GPRS System Option and 4468 EDGE System Option features the following GPRS menus described in the respective sections:

- "GPRS basic menu" – for quick overall tests of GPRS mobile phones
- "GPRS phase/frequency menu" – for transmitter tests of phase and frequency behavior
- "GPRS burst menu" and "GPRS Burst Zoom Menu" – for the burst and power/time template measurements
- "GPRS receiver menus" – for the quality tests of the mobile's receiver
- "GPRS parameter menus" – for setting the system parameters

To switch between different GPRS menus, use the marker tabs on the right-hand side of the GPRS Basic menu.

GPRS basic menu

The GPRS Basic menu enables you to test the mobile's most important characteristics in synchronous (call) mode, in which the mobile is GPRS attached and DL/UL TBFs can be established. The GPRS Basic menu allows setting the test parameters and provides a quick overall view over the measurement result. Using the GPRS Basic menu, you can select from a variety of tests, parameters and measurement output displays.
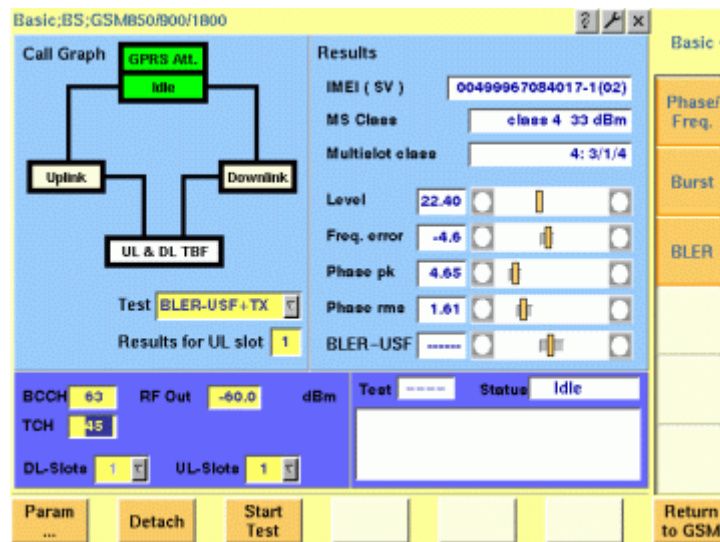
The GPRS Basic menu provides the following entry and output fields:

–   Graphic overview of the ongoing signaling
    In addition to information described in "Call graph" on page 41, the call
    graph provides an additional GPRS Attach field to indicate whether the
    mobile under test is GPRS attached:

    –   The GPRS Attach field signals by white background that GPRS data
        transfer is supported but the mobile is not GPRS attached.

    –   The GPRS Attach field signals by yellow background that the mobile is
        GPRS attached.

–   Test Parameters entry field area for input parameters

–   IMEI (international mobile equipment identity) and MS Class (power class)
    of the mobile under test

–   Multislot class
    The GPRS multislot class of the mobile under test is displayed here in the
    format n: x/y/z, where n = 1...29 is the GPRS multislot class, x is the
    maximum number of downlink slots, y is the maximum number of uplink
    slots, and z is the maximum sum of downlink and uplink slots that can be
    transmitted simultaneously.

–   Basic transmitter and receiver measurement results

Softkeys

The following softkeys are available on the GPRS Basic menu:

–   **Param...** softkey – provides access to GPRS Parameter menus.

–   **Detach** softkey – allows you to invoke a base station-originated GPRS
    detach once the mobile has attached to the simulated network.

–   **Start Test** softkey – starts a test.

–   **Return to GSM** softkey – quits the GPRS Basic menu and returns to the
    GSM Basic menu.

Input parameters

For test parameters to be entered before starting a test, see "Test parameters" on
page 123.

Results

For basic information on results provided on the GPRS Basic menu, see "Basic
test results display" on page 43. The results displayed on the GPRS Basic menu
apply to the slot set by the input parameter Results for UL slot.

Tool bar

A tool bar is accessible from the GPRS Basic menu. It gives access to additional
4463 GSM/GPRS System Option and 4468 EDGE System Option's functionality
outside the system application. The GPRS synchronous mode tool bar includes
the following features:

–   Spectrum

–   Utilities

–   Config

To access the GPRS synchronous mode tool bar, push the GSM/GPRS marker tab on the Welcome menu, then push the **Calls & Meas.** softkey, subsequently push the **GPRS...** softkey and finally the Tools key (?) . The tool bar will appear at the right-hand side of the menu, providing access to subsequent menus.

Starting a test

To start a test, push the **Start Test** softkey on the GPRS Basic menu. The GPRS Basic menu with the softkey label changed to **Stop Test** will appear.

Using the softkeys, you can proceed as follows:

- To access the GPRS Parameter menus, press the **Param...** softkey. This allows you to view but not to change the current parameters.

- To freeze the display, press the **Freeze** softkey.

- To reset the statistics, press the **Reset** softkey.

- To stop the test, press the **Stop Test** softkey.

You can switch between different measurements using the Test Mode selection field. This stops the current test and starts a new test scenario.

> **NOTE**
>
> During a test, all entry fields can be modified except BCCH, PDTCH and MS Pwr.

GPRS phase/frequency menu

The GPRS Phase/Frequency menu enables you to evaluate the modulation quality of the mobile under test. For basic information on phase/frequency measurements in synchronous mode, see "Phase/Frequency test menu" on page 45. To access the GPRS Phase/Frequency menu, press the Phase/Freq. marker tab on the GPRS Basic menu.
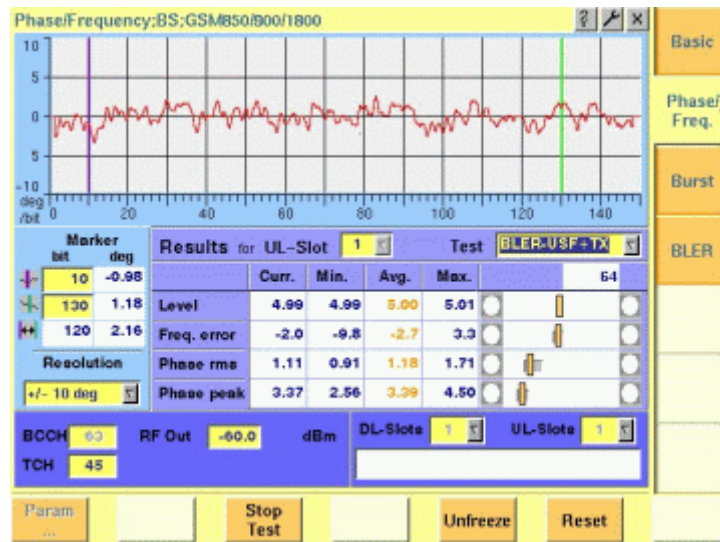
Input parameters

For test parameters to be entered before starting a test, see "Test parameters" on page 123.

Results

For basic information on results provided on the GPRS Phase/Frequency menu, see "Phase/Frequency test menu" on page 45. The results displayed on the GPRS Basic menu apply to the slot set by the input parameter Results for UL-Slot.

The Phase/Frequency Live display shows the phase deviations versus time over the duration of the selected burst. The display markers make the phase error reading at any position of the Phase/Frequency Live display easy. The output fields below the graphic representation display the statistical values and the position of the test results relative to the measurements limits. The numerical outputs of the GPRS Phase/Frequency menu include:

– Level – the RF power level received by the 4463 GSM/GPRS System Option and 4468 EDGE System Option (in dBm)

– Freq. error – the mobile's carrier offset relative to the expected frequency (in Hz)

– Phase rms – the mean difference between the modulated signal and the perfect signal (in degrees, rms-valued)

– Phase peak – the maximum phase error detected within the selected burst (in degrees)

GPRS burst menu   The GPRS Burst menu allows a lot of measurements to determine the burst quality of the mobile under test. For basic information on burst measurements in synchronous mode, see "Burst test menu" on page 47. To access the GPRS Burst menu, press the Burst marker tab on the GPRS Basic menu.
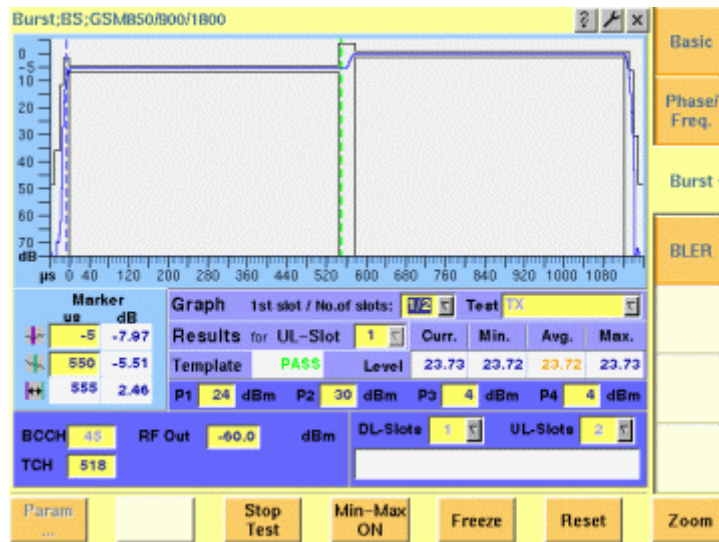
Input parameters

For test parameters to be entered before starting a test, see "Test parameters" on page 123.

Results

The measurement results are displayed in the output fields given below.

Template — Indicates whether the mobile under test satisfies or violates the PTT limits for the multislot operation. The message FAIL is displayed if a measurement in any time slot involved in the test violates the PTT limits. The message PASS is displayed if a measurement in all time slots fulfills the PTT limits.

Level — Display the results of the statistical evaluation of the power level measurements in the time slot set by the input parameter UL-Slot.

GPRS Burst Zoom Menu

To access the GPRS Burst Zoom menu, press the **Zoom** softkey on the GPRS Burst menu. Two different menus are accessible on the GPRS Burst Zoom menu via marker tabs:

– To display the power transition phase of the burst, press the RAMP marker tab.
  This menu provides you with specific functions to investigate both the rising and the falling edge of the burst in detail.
– To display the constant power phase of the burst, press the Constant Power marker tab.
  This menu enables you to check all details of the part of the burst that should have a constant power level.

The figure below shows the zoom display of the burst's power transition phase.
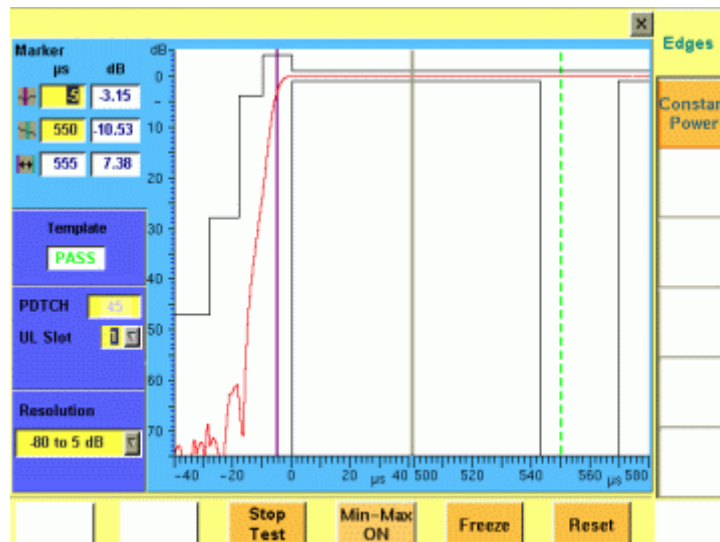
Input parameters

For test parameters to be entered before starting a test, see "Test parameters" on page 123.

Results

The GPRS Burst Zoom menu displays graphically the selected phase of the burst set by the input parameters on the GPRS Burst menu.

The output field Template indicates whether the mobile under test satisfies or violates the PTT limits for the multislot operation. The message FAIL is displayed if a measurement in any time slot involved in the test violates the PTT limits. The message PASS is displayed if measurements in all time slots fulfill the PTT limits.

GPRS receiver menus

The GPRS Receiver menus allow a variety of measurements to evaluate the quality of the mobile's transmitter and receiver in different test modes (see Section , "GPRS test modes").

To access a GPRS Receiver menu, press the BLER marker tab on the GPRS Basic menu. To select the test mode, choose an item listed in the Test Mode entry field. The relevant GPRS Receiver menu will appear. The table below summarizes the features of the test modes implemented by the 4463 GSM/GPRS System Option and 4468 EDGE System Option.

Test Mode	Characteristic	Results
TX	Fast TX measurements	TX results per slot
BLER–BCS	RX on up to 4 slots	BLER–BCS
BLER–USF+TX	Slow RX+TX measurements	BLER–USF per slot TX measurements

NOTE

In TX test mode, the menu does not display any input or result fields.

BLER–BCS   The figure below shows the GPRS Receiver menu for the BLER-BCS test mode.
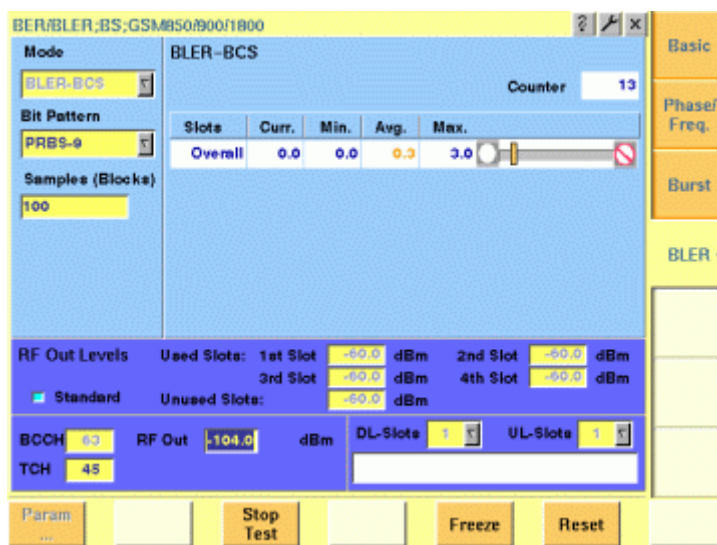
Input parameters

For test parameters to be entered before starting a test, see "Test parameters" on page 123.

Results

The results of the BLER-BCS measurements are displayed in the output fields given below.

Counter — Number of the sample (block) currently used for the statistical evaluation

Curr. — BLER-BCS value of the current test run

Min. — Minimum BLER-BCS value of all test runs

Avg. — Average BLER-BCS value of all test runs

Max — Maximum BLER-BCS value of all test runs

BLER–USF+TX   The figure below shows the GPRS Receiver menu for the BLER-USF+TX test mode.
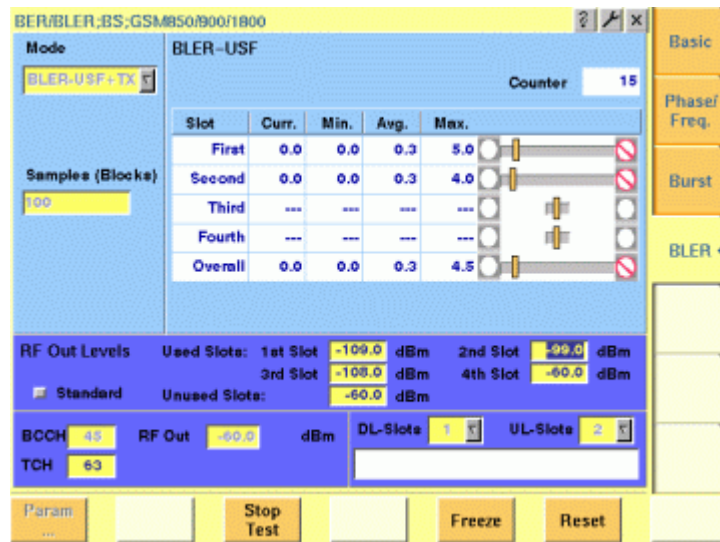
Input parameters

For test parameters to be entered before starting a test, see "Test parameters" on page 123.

Results

The results of the BLER-USF+TX measurements are displayed in the output fields given below.

Counter — Number of the sample (block) currently used for the statistical evaluation

Slot — Number of the downlink slot for which the results are displayed
Overall applies to overall performance over all used slots

Curr. — BLER-USF value of the current test run

Min. — Minimum BLER-USF value of all test runs

Avg. — Average BLER-USF value of all test runs

Max — Maximum BLER-USF value of all test runs

Test parameters In this section, the test parameters to be entered before starting a test are summarized for all GPRS menus.

BCCH — Channel number of the base channel in the time slot 0. Valid range:
1...124 for P-GSM 900
0...124, 975...1023 for E-GSM 900
0...124, 955...1023 for R-GSM 900
512...885 for GSM 1800
512...810 for GSM 1900
128...251 for GSM 850
Default is 63.

NOTE

The value of the BCCH carrier cannot be changed during an active UL or DL TBF.

Bit Pattern — Entry field to set the bit sequence used for the BLER-BCS measurement. Valid entries are:
0000... (the Willtek 4400 transmits all zeros)
1111... (the Willtek 4400 transmits all ones)
1010... (the Willtek 4400 transmits a sequence of bit reversals)
0101... (the Willtek 4400 transmits a sequence of bit reversals)
PRBS-9 (the Willtek 4400 transmits pseudorandom bit sequence with 2^9-1 different states)
PRBS-15 (the Willtek 4400 transmits pseudorandom bit sequence with 2^{15}-1 different states)
PRBS-23 (the Willtek 4400 transmits pseudorandom bit sequence with 2^{23}-1 different states)

NOTE

This entry field is not available in the BLER-USF+TX test mode.

1st slot / No.of slots — Entry field to set which of the sequence of bursts shall be displayed. Valid entries depend on the parameter No. UL-Slots, default is 1/1. The first number designates which of the active time slots the Willtek 4400 shall begin with on the Burst live display.
The second number defines how many from the sequence of bursts shall be displayed.

Marker — Entry fields to set the display markers for reading of the measurement results at specific positions.

No. DL-Slots — Number of PDTCHs to be transmitted by the Willtek 4463 GSM/GPRS System Option and 4468 EDGE System Option. No. DL-Slots successive downlink slots are assigned by the 4463 GSM/GPRS System Option and 4468 EDGE System Option depending on the mobile's capabilities. The value can be entered after the mobile has been GPRS Attached and after it has informed the Willtek 4400 of its multislot class. The value is reset to 1 after a GPRS Detach. Note: The number of downlink slots is selectable only after a GPRS Attach, as the mobile informs the 4400 of its multislot class during the GPRS Attach procedure. The valid range depends on the multislot class n: x/y/z of the mobile under test displayed in the Multislot class field (see Section , "GPRS basic menu"). No. DL-Slots must be less than or equal to the parameter x, and the sum of No. DL-Slots and No. UL-Slots must not exceed the parameter z.

No. UL-Slots — Number of physical channels to be used by the mobile station. The value is automatically adjusted when a mobile registers and announces a multislot class supporting less channels than defined in this field. No. UL-Slots successive uplink slots are assigned by the 4463 GSM/GPRS System Option and 4468 EDGE System Option depending on the mobile's capabilities.
Note: The number of uplink slots is selectable only after a GPRS Attach, as the mobile informs the 4400 of its multislot class during the GPRS Attach procedure. The valid range depends on the multislot class n: x/y/z of the mobile under test

displayed in the Multislot class field (see "GPRS basic menu" on page 116). No. UL–Slots must be less than or equal to the parameter y, and the sum of No. DL–Slots and No. UL–Slots must not exceed the parameter z.

TCH — Packet data traffic channel number. Valid range:
1...124 for P-GSM 900
0...124, 975...1023 for E-GSM 900
0...124, 955...1023 for R-GSM 900
512...885 for GSM 1800
512...810 for GSM 1900
128...251 for GSM 850
Default is 63.

NOTE

The value of the TCH carrier can be changed during an active UL and/or DL TBF.

P1, P2, P3, P4 — Four entry fields to set the power levels $Pn = Gamma_0 - Gamma_{CH}$ for the uplink slots, where n is the number of the active time slot within a TDMA frame, for mobile station output power control according to GSM 05.08 (see "Transmit Power" on page 111).

NOTE

The power levels can also be changed during an active TBF.

Resolution — Entry field to set the scale of the vertical axis (the phase error). The possible settings are ±2, ±5, ±10, ±20 and ±50 degrees, default is ±10 degrees.

Results for UL–Slot — Number of the slot (burst) for which results are displayed. Valid range is 1...(No. UL–Slots), default is 2.

RF Out — Output level of the 4463 GSM/GPRS System Option and 4468 EDGE System Option in case that all time slots use the same power level. Valid range is –120...–10 dBm.
Note: This entry field is available only if the RF Out Levels Standard check box on GPRS Receiver menus is selected.

RF Out Levels Standard — Check box to activate or deactivate the input fields for downlink power levels. If this check box is selected, all time slots use the same power level set by RF Out. Default is "on".

RF Out Levels Unused Slots — Entry field for the power levels in the unused time slots, excluding BCCH. (The BCCH power level is selected by RF Out.)
Note: This entry field is available only if the RF Out Levels Standard check box is not selected.

RF Out Levels Used Slots — Four entry fields for the power levels in the up to four usable time slots.
Default is –60 dBm.
Note: This entry field is available only if the RF Out Levels Standard check box is not selected.

Samples — In case of BLER measurements, this is the number of blocks used to get one measurement result back. In the BLER-BCS case, the number of samples refers to one time slot, so the measurement always takes the same time no matter on how many time slots the BLER-BCS is being performed.
Valid range: 1...10,000.
Default is 100.

Test Mode — Entry field to select the type of measurement. Valid entries are TX, BLER-BCS, BLER-USF+TX, default is BLER-USF+TX.

Upper GSM band — This parameter can be used to switch between GSM 1800 and GSM 1900. It affects only channel numbers in these bands (512...885 for GSM 1800 and 512...810 for GSM 1900). If the band is changed, the BCCH and TCH values should be checked. Valid entries are 1800, 1900, default is 1800.

GPRS parameter menus

The GPRS Parameter menus allow you to set a number of system parameters including the power/time templates, test limits, the coupling loss curve and the corner points. For basic information on system parameters, see "System parameters" on page 76. To access a GPRS Parameter menu, press the **Param...** softkey while working at the test level of any GPRS test.

NOTE

During a test, the GPRS Parameter menus are accessible but the parameters cannot be changed.

There are six different GPRS Parameter menus selectable by the respective marker tabs:

– Basic – see "GPRS basic parameter menu" on page 126
– PTT – see "GPRS PTT parameter menu" on page 127
– Limits –see "Measurement limits" on page 85
– Power Limits – see "Power measurement limits" on page 88
– Coupling Loss – see "Coupling loss" on page 90
– Corner Points – see "Corner points" on page 94

GPRS basic parameter menu

The GPRS Basic Parameter menu allows you to set the parameters of the base station simulated by the 4463 GSM/GPRS System Option and 4468 EDGE System Option in GPRS synchronous test mode.
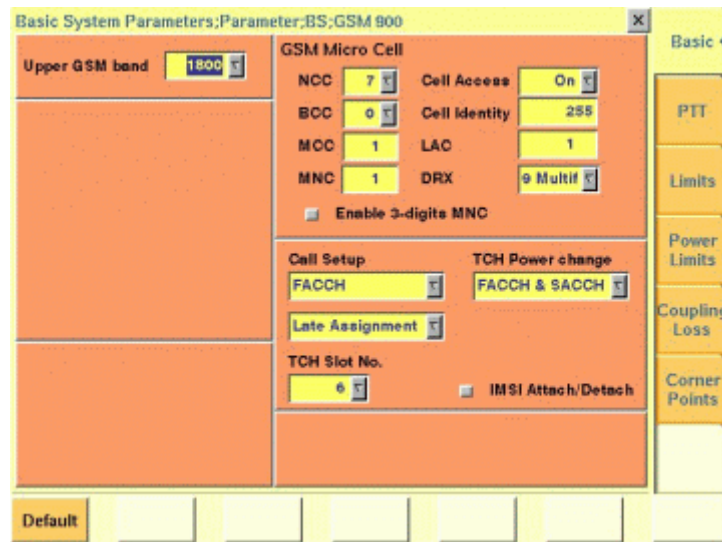
Input parameters

The GPRS Basic Parameter menu is divided into several areas. The parameter fields in the GSM Micro Cell Simulation area are identical to those in GSM environment (see "Basic system parameters" on page 76). For other input parameters, see "Test parameters" on page 68.

Reset

To reset all parameters in the 4463 GSM/GPRS System Option and 4468 EDGE System Option GSM/GPRS System Option to default values, press the **Default** softkey.

> **NOTE**
>
> Pressing the **Default** softkey will delete all values entered on all GSM/GPRS parameter menus and set them to factory defaults.

Help

To access the Help menu, press the Help key ` ? ` .

GPRS PTT parameter menu

In GPRS test mode, there are three menus for setting PTT parameters. To access the PTT Normal menu, the PTT Paging or the PTT Trans menu, press the **Normal**, **Paging** or **Trans.** softkeys on the GPRS PTT Parameter menu, respectively. For basic information on power/time templates and how to set the PTT limits, see "Power/time templates (PTT)" on page 81.

PTT Normal Menu

The upper and lower limits of the RF power level entered on the PTT Normal menu apply to the transmission of normal bursts, for example on a traffic channel. In multislot operation, the values set for the rising and falling edges on the PTT Normal menu are the limits for the first and the last burst sent by the mobile within a TDMA frame, respectively. The limits for the active (constant) part of the burst apply to each time slot with its respective peak power level.

PTT Trans Menu

The PTT Trans menu allows you to set additional PTT parameters for the transition phase between consecutive bursts.
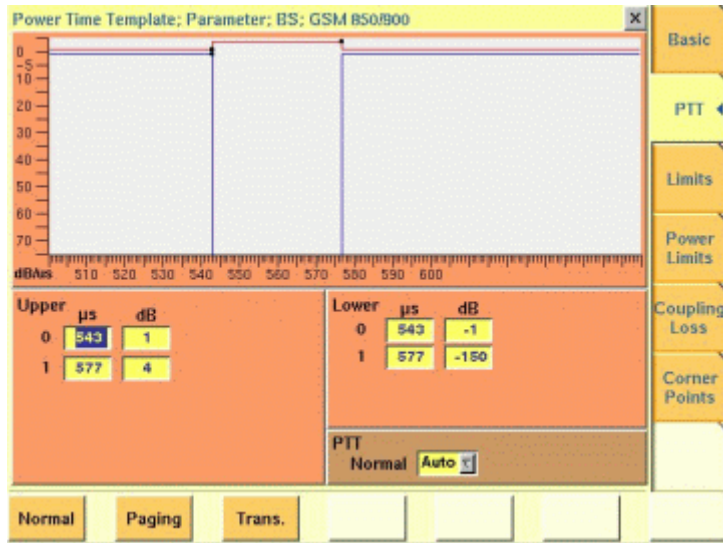
GSM/GPRS Asynchronous Mode Operation

<div style="text-align: right; border: 1px solid black; display: inline-block;">

3

</div>

This chapter provides task-based instructions for using the asynchronous mode (Generator/Analyzer) within the 4463 GSM/GPRS System Option and 4468 EDGE System Option. Topics discussed in this chapter are as follows:

Overview

The generator/analyzer mode of the 4400 provides test and measurement tools for asynchronous measurements; these are measurements without call setup and without signaling.

This section consists of the following parts:

- RF generator – This section describes the menu of the RF signal generator. Using the RF generator, continuous or burst signals can be generated, either unmodulated, with GMSK modulation or with AM.
- Phase/frequency analyzer – Similar to the standard phase and frequency error measurement, the 4400 will analyze the bursts received in terms of phase and frequency error.
- Burst analyzer – Similar to the standard burst measurement, the 4400 will analyze the shape of the bursts received using the standard power/time template.
- Alignment – A predefined series of checks and alignments can be carried out, depending on the type of mobile phone.
- Parameter menus – The most frequently used parameters can be adjusted to your needs in the respective generator or analyzer menu. Some additional parameters are available that affect the generated signal and the measurement views; these parameters can be modified in the parameter menus.

To gain access to the menus of the generator/analyzer tool:

1 Go to the Welcome menu (see picture below).

2 Push the marker tab for the corresponding system option: GSM/GPRS.

3 Push the **Gen/Ana** softkey.

RF generator

To gain access to this menu, push the [RF Gen] marker tab while working with the generator/analyzer tool.

This menu allows to set the parameters of the RF Generator.

The RF generator can be set to generate radio signals from 800 to 2000 MHz; either as continuous or as burst signals.

Modulation can be GMSK – or the signals can be totally unmodulated in order to stimulate pure RF circuitry.

In addition, AM signals can be generated for AM suppression tests. The AM Signal Generator is an option.
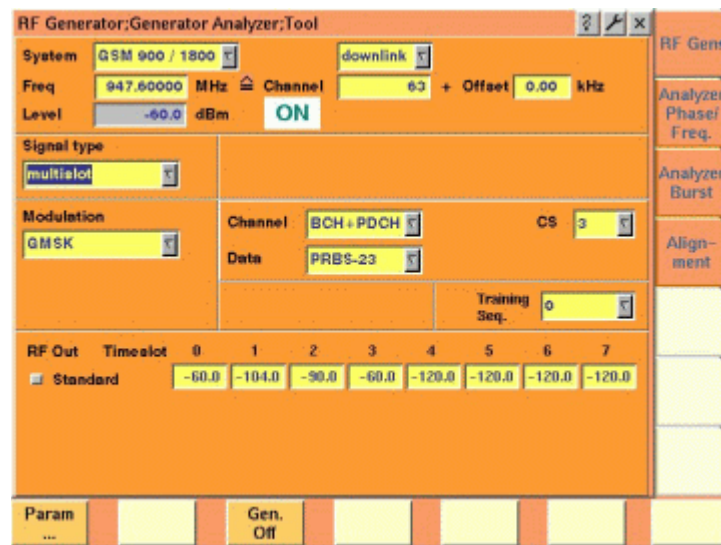
The RF generator menu consists of the following main areas:

– The **system area** allows to set the generator frequency either in MHz or in GSM channel numbers.

– The **signal type area** is where you select bursted or continuous signals.

– The **modulation area** allows to set the modulation parameters.

– The **RF Out area** allows to define individual output levels for the different time slots.

Parameters of the system area

The output frequency of the RF generator and the RF output power level are set in this area.

System — Selection field to set the frequency range of the channel numbers above 511.

Selections possible: GSM 900 / 1800 or GSM 900 / 1900.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

For details regarding GSM frequency ranges, check with section "Basic specifications of GSM bands" on page 97.

uplink / downlink — Selection field to specify either the uplink or the downlink frequency range.
Selections possible: uplink or downlink.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

For details regarding GSM frequency ranges, check with section "Basic specifications of GSM bands" on page 97.

Freq — Entry field to set the output frequency of the RF generator.
Entry range: 800 to 1000 MHz and 1700 to 2000 MHz in increments of 10 Hz.

NOTE

When the frequency entered corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display the channel number and an offset (if any) on the two entry fields of this line.

In case the optional frequency extension has been installed on your 4400, the frequency range available will be different (see specifications of the frequency extension option for details).

Channel — Entry field for the GSM channel number.
Entry range: 0 to 1023

NOTE

When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display the corresponding channel number as the default on this entry field.

Offset — Entry field for the offset from the GSM channel frequency specified.
Entry range: –99.99 to 99.99 kHz in increments of 10 Hz.

NOTE

When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display any offset from the nominal GSM channel frequency as the default on this entry field.

Level — Entry field for the RF output power level of the RF generator. The output signal will be available on the RF in/out connector.
Entry range: –120.0 to –10.0 dBm in increments of 0.1 dB.

> **NOTE**
>
> The level set on this entry field will only be correct when the load impedance is 50 ohms.

> **NOTE**
>
> This entry field is not accessible if individual power levels are set, i.e. if the Signal type is set to multislot and the check box RF Out Standard is selected.

ON / OFF

Display field for the current output status of the RF generator. When ON is displayed, RF signals will be output; when OFF is displayed, the RF generator will be silent.
To toggle between the two states, use the **Gen. Off** respectively the **Gen. On** softkey.

Parameters of the signal type area

The type of the RF signal to be generated is set in this area of the menu.

Signal type — Selects one of the following signal types: continuous (with a constant power level), burst (single-slot) or multislot. The latter can have individual power levels at each of the eight time slots per TDMA frame.

> **NOTE**
>
> Setting "continuous" means that there will be a constant RF output power level (without power ramping). The generated RF signal can still be GMSK modulated (see description of the menu area below).

> **NOTE**
>
> When "continuous" has been selected on this field, the Timeslot entry field will not be available.

> **NOTE**
>
> In case "continuous" has been selected together with GMSK modulation, all time slots will be filled with data and a training sequence (if selected).

Timeslot — Selection field to specify the time slot during which the burst will be generated.
Selections possible: 0 to 6.

> **NOTE**
>
> When "continuous" has been selected on the Signal type entry field, this entry field will not be available.

━━━━ **NOTE**

The internal frame synchronization signal of the 4400 will be available on the **SYNC IN/OUT** connector on the rear of the test set.

Parameters of the modulation area

The modulation of the RF signal to be generated is set in this area of the menu.

Modulation — Selection field.
Selections possible: GMSK, none or AM.

━━━━ **NOTE**

Setting "none" means that the RF output signal generated by the 4400 will be unmodulated, i.e. a pure carrier sine wave.

━━━━ **NOTE**

When "none" has been selected on this field, all other entry fields of this menu area will not be available.

━━━━ **NOTE**

The modulation setting also affects measurements in the analyzer: If GMSK is selected, the analyzer will attempt to demodulate the signal; phase and frequency error are shown with respect to a GMSK-modulated signal. Adversely, if no modulation is selected in the generator, the analyzer shows the results assuming that the input signal is unmodulated as well.

━━━━ **NOTE**

The AM selection is available only if the AM Modulation Option is installed and if Signal type is set to "continuous". For more information regarding AM see below.

Channel — Selects one of the following channel types: GSM, PDCH or BCH+PDCH.
For GSM channel, the 4463 GSM/GPRS System Option and 4468 EDGE System Option generates a modulated signal with a data pattern specified in the Data entry field. If the check box Training Seq. is selected, the data pattern is filled into two blocks of 58 "encrypted bits"; otherwise, the data pattern is filled into all 148 bits of the burst. No channel coding is performed.
Depending on the signal type selected in the Signal type entry field, the signal is sent either continuously (with constant power level or individual power levels) or as bursts. It does not take into account any tail bits or multiframe structures with idle frames.
If PDCH is selected, a signal with one or multiple time slots of channel combination 13 (PDTCH/F + PACCH/F PTCCH/F) is generated. The PDTCH/F blocks can be split into RLC/MAC headers with user-definable contents and a pseudorandom

bit sequence PRBS (selectable in the Data entry field) as data payload. If multislot signal is selected in the Signal type entry field, the time slots of each TDMA frame are filled with exactly the same contents.

NOTE

This entry field is available only for burst or multislot signal type.
The BCH+PDCH channel type is available for the multislot signal type only. At the time slot 0, the 4463 GSM/GPRS System Option and 4468 EDGE System Option transmits a physical channel with channel combination 5 (FCCH + SCH + BCCH + CCCH + SDCCH/4 + SACCH/4). The BCCH includes transmission of System Information Message Type 13, which relays information about GPRS capabilities of the simulated network. The bit pattern selectable in the Data entry field does not apply to the time slot 0. The other time slots (1...7) are filled with identical PDCH data, see description of PDCH above.

NOTE

Some of the parameters transmitted in the System Information Messages on the BCCH can be affected on the Basic Parameter menu.

CS — Selects the coding sequence in the range 1...4 if PDCH or BCH+PDCH is selected in the Channel entry field. This parameter refers to the coding algorithm on the PDTCH.

Data — Selects the data payload for GSM channels (uncoded) and packet data traffic channels.
For GSM channel selected in the Channel entry field, valid entries are 0...0, 1...1, 1010..., 1100..., 11110000, 1...10...0, PRBS-9, PRBS-15, PRBS-23.
If PDCH or BCH+PDCH is selected in the Channel entry field, valid entries are PRBS-9, PRBS-15, PRBS-23.

Training Seq. — Check box and selection field.
With the check box, transmission of a training sequence in each time slot can either be selected or deselected.
If transmission of a training sequence has been selected, the training sequence code can be chosen (range of the selection field: from 0 to 7).

NOTE

Each one of the eight training sequence codes refers to a predefined 26-bit training sequence for normal bursts.

NOTE

When a training sequence has been selected, it will be transmitted in the middle of every time slot. Thus, there will be some kind of time slot structure of the RF signal generated, even when "continuous" has been selected as the Signal type.

NOTE

However, there will be no GSM frame structure as the RF Generator will not provide a BCCH (and thus, there will be no mark for a slot '0' apart from the internal frame synchronization signal of the 4400, which is available on the **SYNC IN/OUT** connector on the rear of the test set).

RF Out Standard — If selected, the 4463 GSM/GPRS System Option and 4468 EDGE System Option will apply the RF output level entered in the Level field to all time slots, and the RF Out Timeslot entry field for individual time slots will not be accessible.

> **NOTE**
>
> This check box is available only for the multislot signal type.

RF Out Timeslot — Allows setting the RF power level of each time slot (0...7) if the RF Out Standard check box is not selected. For each time slot, any RF power level within the specified range (usually –120...–10 dBm) is permissible.
The following softkeys are available on the Generator menu:

- The **Gen On/Gen Off** softkey turns the generator on and off.
  After pressing the **Gen On** softkey, the 4463 GSM/GPRS System Option and 4468 EDGE System Option transmits the signal configured, and the softkey label turns to **Gen Off**. Pressing the **Gen Off** softkey turns the generator off and changes the softkey label to **Gen On**.

- The **Param...** softkey gives access to Parameter menus

AM Modulation

AM Modulation is an option to the 4400 and the functionality is available only if installed. To check if the option is installed, see under **TOOLS > Config > Options**.
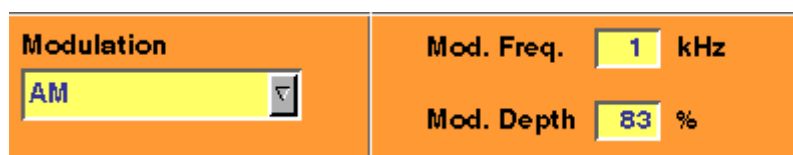
A few manufacturers recommend to use an AM modulated signal with a defined modulation frequency and depth to test the mobile's capability to reject AM signals.

AM modulation is selectable only if the signal type is "continuous". The carrier can be modulated with a definable sine wave.

If AM modulation is selected, the Data and other fields disappear and are replaced by input fields for the AM parameters.

Mod. Freq. — The modulation frequency is modulated onto the carrier frequency.
Valid entries are in the range from 0 to 99 kHz in steps of 1%.

Mod. Depth — The modulation depth describes the degree to which the carrier is turned low by the negative peak.
Valid entries range from 0 to 99% in steps of 1%.

Phase/frequency analyzer

To gain access to this menu, push the [Analyzer Phase/Freq.] marker tab while working with the generator/analyzer tool.
This menu displays the results of the phase and frequency error measurements of the RF signal received at the **RF IN/OUT** connector.

> **NOTE**
>
> The analyzer will measure the signal and display the results assuming that the signal is GMSK-modulated if the 4400 generator is set to GMSK as well. If the generator of the 4400 is set to transmit an unmodulated signal, the measurements in the analyzer are calculated assuming that the signal at the RF input connector is unmodulated as well.
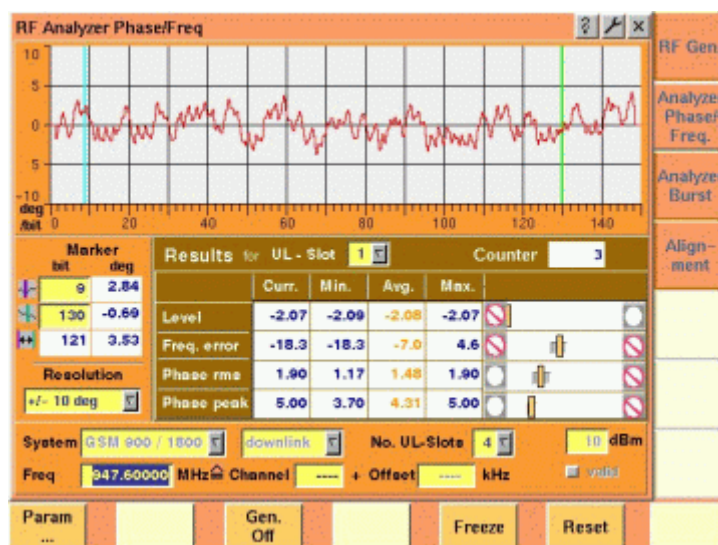
Using the phase/frequency analyzer

The RF analyzer takes measurements in asynchronous mode, without any call processing functions.
Any Coupling loss table defined in the appropriate menu will be used for asynchronous measurements as well.
Before starting a measurement, the radio frequency to be measured on has to be entered.
There are entry fields available for these parameters in the lower area of the menu, just above the softkeys.
The measurement results, resolution settings, and display markers are similar to the corresponding areas and fields of the standard GSM test menu.

Parameter fields of the RF analyzer

The radio frequency to measure on and the trigger functions are set in this menu area.

System — Selection field to set the frequency range of the channel numbers above 511.
Selections possible: GSM 900 / 1800 or GSM 900 / 1900.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

NOTE

For details regarding GSM frequency ranges, check with section "Basic specifications of GSM bands" on page 97.

uplink / downlink — Selection field to specify either the uplink or the downlink frequency range.
Selections possible: uplink or downlink.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

NOTE

For details regarding GSM frequency ranges, check with section "Basic specifications of GSM bands" on page 97.

Freq — Entry field to set the measurement frequency for the RF analyzer.
Entry range: 800 to 1000 MHz and 1700 to 2000 MHz in increments of 10 Hz.

NOTE

When the frequency entered corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display the channel number and an offset (if any) on the two entry fields of this line.

NOTE

In case the optional frequency extension has been installed on your 4400, the frequency range available will be different (see specifications of the frequency extension option for details).

Channel — Entry field for the GSM channel number.
Entry range: 0 to 1023

NOTE

When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display the corresponding channel number as the default on this entry field.

Offset — Entry field for the offset from the GSM channel frequency specified. Entry range: –99.99 to 99.99 kHz in increments of 10 Hz.

> **NOTE**
>
> When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display any offset from the nominal GSM channel frequency as the default on this entry field.

No. UL–Slots — Selects the expected number of uplink time slots in the range 1...4.

Result for UL–Slot — Selects an uplink time slot within 1...4 to display numerical results.

Softkeys of the RF Analyzer

Param ...	This softkey leads you to the parameter screens where you can set up coupling loss factors and limits.
Gen. Off	A push on this softkey switches the RF generator either on or off.
Freeze	To freeze the current display for further or detailed analysis (or for printing), push this softkey. The softkey will change to **Unfreeze**. **Note:** After the **Freeze** softkey has been pushed, all measurements will continue in the background, but the display will not be updated.
Reset	A push on this softkey resets the statistic evaluation of the measurement results. The measurement counter will be reset as well.

Burst analyzer

To gain access to this menu, push the [Analyzer Burst] marker tab while working with the generator/analyzer tool.
This menu displays the results of the asynchronous burst measurements of the RF signal received at the RF in/out connector.

> **NOTE**
>
> The analyzer will measure the signal and display the results assuming that the signal is GMSK-modulated if the 4400 generator is set to GMSK as well. If the generator of the 4400 is set to transmit an unmodulated signal, the measurements in the analyzer are calculated assuming that the signal at the RF input connector is unmodulated as well.

Using the burst analyzer

The RF analyzer takes measurements in asynchronous mode, without any call processing functions.

Any Coupling loss table defined in the appropriate menu will be used for asynchronous measurements as well.

Before starting a measurement, the radio frequency to be measured on has to be entered.

There are entry fields available for these parameters in the lower area of the menu, just above the softkeys.

The measurement results and the display markers are similar to the corresponding areas and fields of the standard GSM test menu.

The graphic display shows the measured RF power versus time, relative to the average power level in the displayed time slot with the highest power. The graph of the power level as a function of time can be shown for one or more time slots. From the up to four time slots that can be received by the 4463 GSM/GPRS System Option and 4468 EDGE System Option, any combination of consecutive bursts can be measured using the First Burst / Number of Bursts entry field.

Parameter fields of the RF analyzer

The radio frequency to measure on and the trigger functions are set in this menu area.

System — Selection field to set the frequency range of the channel numbers above 511.

Selections possible: GSM 900 / 1800 or GSM 900 / 1900.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

NOTE

For details regarding GSM frequency ranges, check with section **"Basic speci-fications of GSM bands" on page 97**.

uplink / downlink — Selection field to specify either the uplink or the downlink frequency range.
Selections possible: uplink or downlink.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

NOTE

For details regarding GSM frequency ranges, check with section **"Basic speci-fications of GSM bands" on page 97**.

Freq — Entry field to set the measurement frequency for the RF analyzer.
Entry range: 800 to 1000 MHz and 1700 to 2000 MHz in increments of 10 Hz.

NOTE

When the frequency entered corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display the channel number and an offset (if any) on the two entry fields of this line.

NOTE

In case the optional frequency extension has been installed on your 4400, the frequency range available will be different (see specifications of the frequency extension option for details).

Channel — Entry field for the GSM channel number.
Entry range: 0 to 1023

NOTE

When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings of the two selec-tion fields above this entry field), the 4400 will automatically display the cor-responding channel number as the default on this entry field.

Offset — Entry field for the offset from the GSM channel frequency specified.
Entry range: –99.99 kHz to 99.99 kHz in increments of 0.01 kHz.

NOTE

When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings on the two selec-tion fields above this entry field), the 4400 will automatically display any off-set from the nominal GSM channel frequency as the default on this entry field.

No. UL–Slots — Selects the expected number of uplink time slots in the range 1...4.

Result for UL–Slot — Selects an uplink time slot within 1...4 to display numerical results.

First Burst / Number of Bursts — The first number and the second number in this entry field indicate the first burst displayed and the total number of bursts displayed respectively.

Softkeys of the RF Analyzer

Param ... — This softkey leads you to the parameter screens where you can set up coupling loss factors and limits.

Gen. Off — A push on this softkey switches the RF generator either on or off.

Freeze — To freeze the current display for further or detailed analysis (or for printing), push this softkey.
The softkey will change to **Unfreeze**.
Note: After the **Freeze** softkey has been pushed, all measurements will continue in the background, but the display will not be updated.

Reset — A push on this softkey resets the statistic evaluation of the measurement results. The measurement counter will be reset as well.

Alignment

The Alignment screen allows you to interactively run a sequence of checks and alignments on the mobile phone in asynchronous mode. An individual sequence can be defined for each type of mobile phone; the phone type is easily selectable. For a selected phone, each step can be carried out on user request, either in the given sequence or out of sequence. Each step can consist of a continuous or burst signal being sent to the phone for stimulation, and/or a signal from the phone being measured in terms of power or spectrum.

This section is divided into two subsections. The first one shows you how to use the Alignment screen; so this is what you need to know when testing a phone. The second one explains you how to set up your own mobile phone scripts. This subsection is applicable to those who want to write a new script, for example for a new type of mobile phone.

Application

Many mobile phones are checked, repaired and aligned according to the manufacturer's test procedures. These procedures consist of test steps typically involving stimulation of the mobile phone with a modulated or unmodulated signal. In most cases, the RF power or the modulation spectrum is measured at

a test point, and compared with given limits. For modulator alignment purposes, the RF power at 67 kHz is often compared with the power at the nominal carrier frequency.

For these purposes, the 4400 provides a screen combining signal generator, power and spectrum analyzer functions with the scripting functionality to give a powerful tool for quick and precise tests and alignments.

The Alignment screen allows you to select a script for a mobile phone. In each script, a number of test steps are defined – just as in the service manual for the respective phone. These test steps can be executed in sequence; alternatively, they can also be performed individually as required.

Each test step can be selected by name; ideally, this name is identical with the name of the test step in the service manual.

Alignment screen elements

As you can see in the picture above, the Alignment screen is divided into three sections:

– The upper half of the screen shows the RF spectrum. Results are only shown or updated when required – in some measurement steps, you will only need the RF power reading.

– The spectrum display contains three markers; the marker section in the lower left part of the screen allows you to adjust their respective horizontal positions and to read the results.

– The lower right section allows you to select the type of mobile phone to be tested and to choose a test step. It also displays the most important RF settings and shows the measured RF power.

Selecting a mobile phone and test step

Two scroll fields allow you to select the type of mobile phone under test and the next test step for the selected phone, respectively.

Type — This scroll field allows you to select a mobile phone. The 4400 is delivered with a few examples; scripts for more mobile phones may be available from the mobile phone manufacturers, from Willtek's support team or from Willtek's web pages.

Step — For each selectable mobile phone, a number of test steps are available. A new step can be selected by its name from the Step scroll box. In each new step, generator and analyzer settings may change as well as the type of measurement. The most important generator (Gen) and analyzer (Ana) settings are shown below the Step scroll box.
Instead of selecting a step with the scroll box, you can select a new step with the softkeys (see below).

Results and Markers

The **Spectrum** display can show you the spectrum within 100 kHz around the center frequency. The result is displayed relative to the peak; this means that the highest point in the frequency spectrum is always 0 dB.
The center channel and frequency are indicated below the selection fields for phone type and test step (Ana TCH, Freq). The spectrum is measured in a 10 kHz resolution bandwidth.

Marker kHz		dBm
▽	-100.0	---
▽	0.0	---
▽	100.0	---
↔	100.0	---
↔	200.0	---
↔	100.0	---

Three markers are available, allowing you to read the numerical results as absolute values and relative to each other.

Many tests require a reading of the **RF power**. The RF power (burst peak power) is shown on the right-hand side, measured in dBm.

A measurement counter indicates how many measurements have been performed and displayed already since the start of the test. This can be useful in order to verify that measurements are still being performed. It can also be used to monitor the progress when averaging results; the averaging functions can be selected by softkey (see below).
The counter and all measurement results are reset with the Reset softkey.

Softkeys on the alignment screen

The softkeys provide different functions to control the functions and the result output:

Start Avg. This softkey can be used to start an averaging function over the measurements. It affects both the spectrum measurements and the RF power measurements.

Previous Step

If you want to repeat a test that is one step above the current test (in the list of test steps), a push on this softkey is sufficient to stop the current test and start the wanted test step.
The list is circular: If there is no previous step available because the present step is the first one in the test list, the 4400 continues with the (last) entry at the end of the list.

Repeat Step

This softkey can be used to restart the test step displayed in the Step field. This may be useful after halting a test, moving to another test menu and returning to this menu.

Next Step

For tests in the defined sequence, this softkey serves to stop the current test and start the next one.
The list is circular: If there is no test step following the current one because the end of the test list has been reached, the 4400 continues with the first entry in the test list. This could be used to start a new test with another phone when the previous phone is done.

Freeze

A push on this softkey stops the measurement update, keeping the last result on the display.
The softkey description changes to Unfreeze, allowing you to continue the measurement update after another push on this button.

Reset

The Reset button restarts the ongoing measurements and sets back the measurement counter. If averaging is active, previous results are discarded and the averaging process is restarted from the next new measurement result.

Writing your own alignment script

In each step, the generator and analyzer settings may be changed and a measurement is usually started. All the steps for one type of mobile phone are defined in a script file. This script is run once to initialize the tests when the type of phone is selected, and later on each time a new test step is selected. The script is stored in ASCII format on the hard disk of the 4400; the location is /rapid/ param. The file name should follow the "filename".BAS format, where "filename" consists of at least one character. Ideally, the file name represents the name or designation of the mobile phone.

To create a new script file or modify an existing one, you can either use a standard editor on an external PC or the built-in RAPID! programming environment which includes an editor.

Existing files can be loaded from floppy disk onto the 4400's hard disk. Please ensure that the file is stored in the correct directory!

Alignment scripts basically follow the language rules and conventions as simple RAPID! programs. See section "RAPID! syntax" on page 279 about RAPID!. Some additional rules apply:

– **List of test steps:** The preparation includes the definition of the names of the steps which should appear in the Step scroll box. This definition is given by the `gui "STEP:a,b,c,...,"` statement. a,b,c,… are to be replaced by the text for the different steps (cases).
If you want to change the sequence of steps, just reorder them in the gui statement.

– **Initialization:** An additional initialization phase sets the 4400 to a known state, for example with settings that are common for all the test steps such as the coupling loss. All these preparatory commands, including the list of test steps, are placed in an empty case statement.

Example:
```
case "" ' Elements of the scroll box
gui "STEP:Ch 56 –55dBm,Ch 700 –85dBm,Ch 700 TX-IQ"
scpi ":RFGenerator:STYPe CWAVe"
scpi ":CONFigure:COUPloss:DATA ""3210_1.5_2"", 897.6,
1.5, 1747.8, 4.0"
scpi ":CONFigure:COUPloss:STATe ON"
```

– **Definition of the actual step:** The actual step is a case. The case is handed over to the script in the following statement: `select case cmdline$`. This statement must precede the first case.

– **Test steps:** Each test step is located in a `case` statement. The text in the case statement is the one presented in the Step scroll box. See example below.

– **User instructions** before the start of a measurement should be placed in a message box. Example:
```
msgbox "Settings done","OK"
status = inkeywait
```

– **Progress indication text:** You can display a message, for example to explain the ongoing test step. Example:
```
open "GUI:" as #1
sub gui(cmd$)
    print #1, cmd$
end sub
gui "STEP:Ch 60, –55dBm"
```

– **Setup and measurement commands:** You can issue 4400 SCPI commands by sending them to a device called "SCPI:". Example:
```
open "SCPI:" as #2
sub scpi(cmd$)
    print #2, cmd$
end sub
scpi ":RFGenerator:STYPe CWAVe"
```

Example:

```
rem -------- BigTel 1234 ----------
open "GUI:" as #1
open "SCPI:" as #2

sub gui(cmd$)
print #1, cmd$
end sub
```

```
sub scpi(cmd$)
print #2, cmd$
end sub

rem "This is an automatic sequence to align the BigTel
1234"

select case cmdline$
case "" ' Setting up the scroll box
gui "STEP:Ch 60 -80dBm,Ch 700 -50dBm,Ch 700 TX-IQ,"
scpi ":RFGenerator:STYPe CWAVe"
scpi ":CONFigure:COUPloss:DATA
""BigTel_1234_with_RF_cable"", 902.0, 1.5, 1742.8,
2.0"
scpi ":CONFigure:COUPloss:STATe ON"

case "Ch 60 -80dBm"
scpi ":RFGenerator:GSM:STATe ON"
scpi ":RFG:GSM:MODulation:TSEQuence:STATe OFF"
scpi ":RFGenerator:GSM:FREQuency 947067710"
scpi ":RFGenerator:GSM:LEVel -80"

case "Ch 700 -55dBm"
scpi ":RFGenerator:GSM:STATe ON"
scpi ":RFG:GSM:MODulation:TSEQuence:STATe OFF"
scpi ":RFGenerator:GSM:FREQuency 1842867710"
scpi ":RFGenerator:GSM:LEVel -55"

case "Ch 700 TX-IQ"
scpi ":MEASure:BLOCkdata:MSPectrum"

end select

close #1
close #2

end
```

In the above example, the scroll box allows for three different test steps. In the first two, the generator is set up. In the third step, modulation spectrum measurements are performed and displayed.

Parameter menus

The Parameter menus allow you to access the parameters relevant to asynchronous mode at a more detailed level, which need not be changed in everyday use. To access the Parameter menus, press the **Param...** softkey on the Generator menu and then press the appropriate marker tab. The figure below shows the Basic Parameter menu.
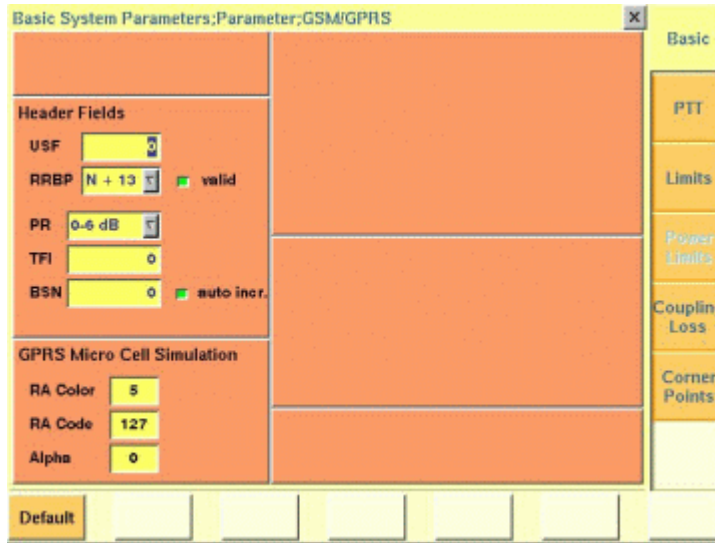
The Basic parameter menu is divided into the following sections:

- The GSM Micro Cell Simulation section provides access to the GSM system parameters broadcast in the System Information messages. The parameter fields are identical to those in GSM environment (see "Basic system parameters" on page 76).

- The GPRS Micro Cell Simulation section provides access to GPRS system parameters broadcast in the System Information messages, especially SI13.

- The Header Fields section defines MAC or RLC header parameter values.

Input Parameters

The input parameters to be entered in the GPRS Micro Cell Simulation and Header Fields sections are listed below.

GPRS Micro Cell Simulation section

RA Color — Sent in the SI3 and SI4 Rest Octets. The normal use is that different different RA Color codes in adjacent cells indicate to the mobile that the cells belong to different Routing Areas. The valid range is 1...7.

RA Code — The Routing Area Code as broadcast in the SI13 Rest Octets. The valid range is 0...255.

Alpha — One of the GPRS Power Control Parameters (SI13 Rest Octets) sent also in the IA Rest Octets and RR Packet Uplink Assignment. Alpha is the coded parameter Alpha involved in the equation below. Valid entries 0...10 correspond to Alpha = 0.0...1.0, default is 10. When the mobile station transmits on a PDCH, it uses the parameter Alpha to determine its output power level Pn of the n-th burst within TDMA frame as follows:

$Pn = min[Gamma_0 - Gamma_{CH} - Alpha(C + 48), P_{max}]$

where $Gamma_0 = 39$ dBm for GSM 900 and $Gamma_0 = 36$ dBm for GSM 1800/1900, $Gamma_{CH}$ is defined for each PDCH individually (e.g. during channel assignment), Alpha is defined above, C is the power level of the basis station received by the mobile, and P_{max} is the maxim output power of the mobile according to its power class.

Header Fields section — The entries in some of the GPRS header fields are user-selectable; for each TDMA frame, one PDTCH time slot is coded and duplicated into all time slots containing the PDTCH.

USF — Defines the Uplink State Flag, which can take on one of eight values either to assign a block to a particular mobile (out of seven possible), or to mark a block as unreserved. It is part of the MAC header on the PDTCH. All time slots use the same USF. The valid range is 0...7.

RRPB — The Relative Reserved Block Period field is part of the MAC header on the PDTCH, taking on one of the following values: N+13, N+17, N+21, or N+26. It marks the block to which the USF field applies relative to the currently used block.

RRBP Valid — This check box enables you to select or deselect the RRBP Valid flag in the MAC header. The flag indicate whether the above RRBP value shall or shall not be used by the mobile station. Default is "off".

PR — The Power Reduction field indicates the RF power in the downlink PDCH relative to the BCCH. It is used in downlink RLC Data Blocks. Valid entries are 0-6 dB, 8-14 dB, 16-22 dB, 24-30 dB, default is 0-6 dB.

TFI — The Temporary Flow Identifier field identifies the Temporary Block Flow TBF to which the RLC data block belongs. This parameter is used in the downlink RLC Data Block. Valid entries are in the range 0...31, default is 15.

BSN — The Block Sequence Number field is the Absolute Block Sequence Number modulo 128 of each RLC data block within the TBF. It is used to enumerate the RLC data blocks, starting with 1. Valid entries are in the range 0...127, default is 1.
This parameter can either be set to a fixed value or be increased by the 4463 GSM/GPRS System Option and 4468 EDGE System Option using the auto incr. check box.

auto incr. — If selected, the BSN is automatically increased with each new RLC data block, starting with a user-defined BSN value. Default is "on".

EDGE Call Mode Operation

4

This chapter describes the functionality of the 4468 EDGE System Option in call mode. Topics discussed in this chapter are as follows:

- "Overview" on page 152
- "EDGE tests" on page 152
- "EGPRS basics" on page 166

Overview

This section covers information on EDGE measurements in synchronous (call) mode. For information on EDGE measurements, see "EDGE tests" on page 152.

EDGE-capable mobile phones are typically tested in standard GSM mode first, and then additional (optionally multislot) measurements in EDGE (EGPRS) mode are made.

To select the synchronous test mode for EDGE, go the Welcome menu (the 4400 start menu), select EGPRS, and press the **Calls & Meas.** softkey.

This system option can test mobile phones designed for the following frequency bands:

- GSM 850
- GSM 900 (P-GSM 900, E-GSM 900, R-GSM 900)
- GSM 1800 (also known as DCS 1800 or PCN)
- GSM 1900 (also known as PCS 1900).

Section "Crash course" on page 4 gives a short guideline on how to connect a mobile and perform tests.
Need to know more about the user interface of the 4400? Just navigate to "Menus and levels" on page 20 and you will find all you need to know.

This chapter is divided into three subsections:

EDGE tests – Here you will find everything regarding the available tests (e.g. Power, EVM/Frequency), how to set up the instrument and how to read the results.

EGPRS basics – If you are not completely familiar with GSM or just need a quick introduction – this is where you will find some explanation of the GSM basics. This chapter also carries many tables on frequency ranges, power classes, burst definitions and so on.

EDGE tests

The 4400 provides you with the following test capabilities:

Basic test menu – For a quick overall test of a mobile

EVM/Frequency menu – TX tests on the modulation quality

EGPRS Burst menu – The burst and power/time template measurements

To switch from one test to another, simply use the marker tabs on the right-hand side of the screen.

> **NOTE**
>
> If a change of the current test menu is not possible (e.g. because a BS call set up has not been completed), the marker tabs will be grayed out. Example: Tests cannot be started and a GPRS Attach cannot be performed until a mobile phone is registered (GPRS-attached).

On every test menu, there are various functions available like Start Test and Detach. These functions can be called by a push on the related softkey below the screen.

The burst test menu provides zooming functions that help you to check the mobile's behaviour in detail. For more information on those zoom levels, please refer to the GSM or GSM/GPRS chapter of this manual.

Basic test menu   The main application of this menu is speeding up the test of the mobile's most important characteristics.
Together with the possibility to adapt the main test parameters to specific requirements and the reduced signaling functionality, this basic test menu is an ideal test-time saver.
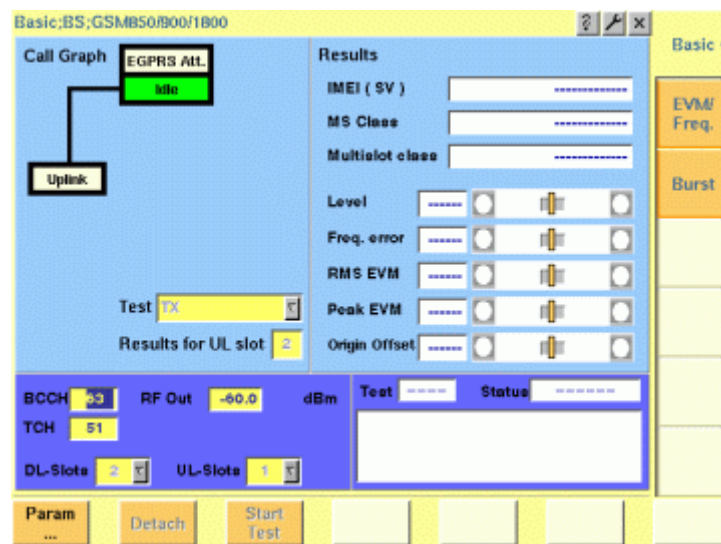
What it does   The basic test menu provides you with

– Graphic overview of the ongoing signaling
   In addition to information described in "Call graph" on page 154, the call graph provides an additional EGPRS Attach field to indicate whether the mobile under test is GPRS attached:

   – The EGPRS Attach field signals by white background that GPRS data transfer is supported but the mobile is not GPRS attached.

   – The EGPRS Attach field signals by green background that the mobile is GPRS attached.

– Test parameters entry field area for input parameters

> – IMEI (international mobile equipment identity) and MS Class (power class) of the mobile under test
>
> – Multislot class
> The GPRS multislot class of the mobile under test is displayed here in the format n: x/y/z, where n = 1...29 is the GPRS multislot class, x is the maximum number of downlink slots, y is the maximum number of uplink slots, and z is the maximum sum of downlink and uplink slots that can be transmitted simultaneously.
>
> – Basic transmitter measurement results

Softkeys   The following softkeys are available on the EGPRS Basic menu:

> – **Param...** softkey – provides access to GPRS Parameter menus.
> – **Detach** softkey – allows you to invoke a base station-originated GPRS detach once the mobile has attached to the simulated network.
> – **Start Test** softkey – starts a test.

Input parameters   For test parameters to be entered before starting a test, see .

Tool bar   A tool bar is accessible from the GPRS Basic menu. It gives access to additional functionality. The GPRS synchronous mode tool bar includes the following features:

> – Spectrum
> – Utilities
> – Audio
> – Config

To access the GPRS synchronous mode tool bar, push the EGPRS marker tab on the Welcome menu, then push the **Calls & Meas.** softkey, and finally push the Tools key. The tool bar will appear at the right-hand side of the menu, providing access to subsequent menus.

The mobile is paged and a GPRS traffic link from the mobile phone to the 4400 is established for test purposes, with the modulation and coding scheme set to MCS-5; this means that the phone shall transmit with 8-PSK modulation.

Call graph   Basically, the call graph is the call status display in graphical form. It allows you to instantly identify the status of a call and the currently executable measurements.
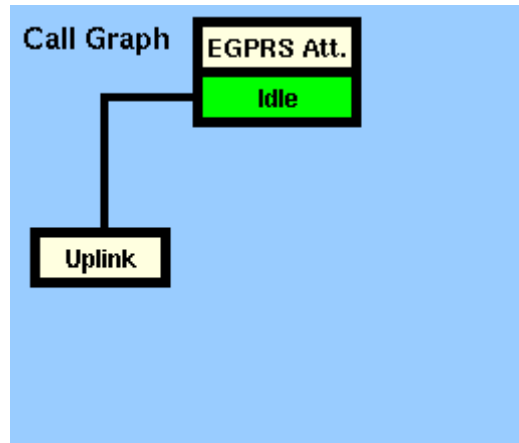
As long as no phone has registered its GPRS capabilities with the 4400 and no paging has been initiated, the call graph will look like the sample above.

How to read the call graph

When the mobile phone has registered its GPRS capabilities with the network simulated by the 4400, the EGPRS Att. field lights green. The field shows a white background as long as there is no mobile phone in the GPRS Attached state.

The rectangular box with the green background indicates the current state of the call.

The orange line with the arrow indicates the path that has been used to achieve the current state.

Uplink active

When the mobile is GPRS attached, a test can be started with a push on the **Start Test** softkey. When an uplink channel has been set up successfully, the Uplink field appears with a green background and the call graph will look as follows:
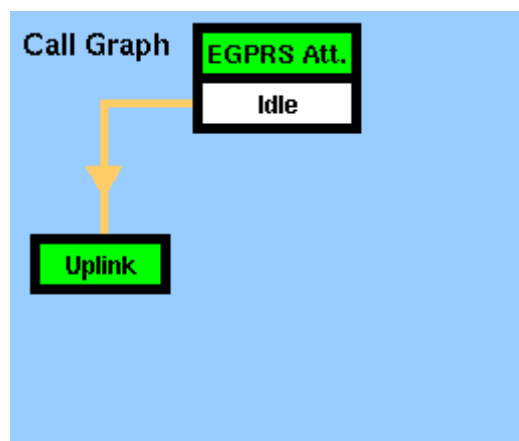
The Uplink status is active now. The EDGE TX measurements will be performed automatically (for details, see section) and the Start Test softkey turns into a Stop Test softkey.

NOTE

If uplink cannot be established, the 4400 will automatically return to idle mode and the Start Test softkey will return.

NOTE

This test will only work with a test SIM being used in the mobile.

Basic test results display The basic test results display gives a short overview of the mobile's identity and transmitter characteristics. Measurement results refer to the selected uplink slot.

How to read the basic test results display

IMEI (SV) Here the 4400 displays the IMEI (= International Mobile Equipment Identity), received from the mobile.
The IMEI is the registration number of the mobile hardware which identifies it internationally. The IMEI is stored inside the mobile's electronics and can only be read out by the 4400 through the radio interface.
The IMEI consists of 15 digits; the last one is not transferred over the radio interface but calculated from the other 14 digits. The additional two digits shown in brackets indicate the software version in the mobile phone which has been transferred to the 4400 together with the IMEI.

MS Class This display field shows the mobile's power class both as GSM power class and as the maximum RF output level of the mobile in dBm.

Multislot class	This display field indicates the GPRS multislot class number together with the maximum number of downlink slots, the maximum number of uplink slots and the maximum total number of slots being used at any one time.
Level	On this bar display the 4400 shows the results of the statistic evaluation of the power level measurements, in dBm. **Note:** The limits for this test depend on the selected slot, the power level chosen for that slot in the burst menu, and the limits selected for that slot in the power limits menu. This menu is a submenu of the Test parameters menu.
Freq. error	Here the 4400 displays the results of the statistic evaluation of the frequency error measurements for the selected slot. The frequency error is shown in Hertz.
RMS EVM	The results of the statistic evaluation of the RMS EVM (error vector magnitude) measurement are displayed on this bar. The error vector magnitude is measured relative to the nominal signal magnitude and shown in %. **Note:** The limits for this test are set on the EGPRS Limits menu.
Peak EVM	On this bar display, the 4400 displays the results of the statistic evaluation of the peak error vector magnitude, in %. **Note:** The limits for this test are set on the EGPRS Limits menu.
Origin offset	Here you will find the results of the statistic evaluation of the origin offset measurement. The origin offset is measured relative to the nominal signal magnitude and shown in %. The limits for this test are set on the EGPRS Limits menu menu.

Starting a test To start a test, push the **Start Test** softkey on the GPRS Basic menu. The GPRS Basic menu with the softkey label changed to **Stop Test** will appear.

Using the softkeys, you can proceed as follows:

– To access the GPRS Parameter menus, press the **Param...** softkey.
  This allows you to view but not to change the current parameters.

– To freeze the display, press the **Freeze** softkey.

– To reset the statistics, press the **Reset** softkey.

– To stop the test, press the **Stop Test** softkey.

You can switch between different measurements using the Test Mode selection field. This stops the current test and starts a new test scenario.

NOTE

During a test, the output power of the 4400 (RF Out) and the packet data traffic channel (TCH) can be changed, as well as the uplink slot for which the results are shown.

EVM/Frequency menu

The EVM/Frequency menu enables you to evaluate the modulation quality of the mobile under test. To access the GPRS Phase/Frequency menu, press the [EVM/Freq.] marker tab within the EGPRS call mode menus.
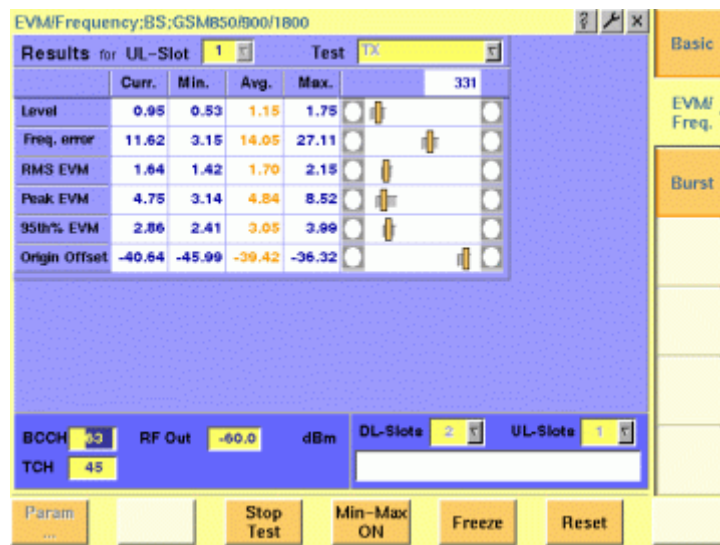
Input parameters

For test parameters to be entered before starting a test, see "Test parameters" on page 161.

Results

For basic information on results provided on the EVM/Frequency menu, see "Required tests" on page 167. The results displayed on the EGPRS Basic menu apply to the slot set by the input parameter Results for UL Slot.

The EVM/Frequency display shows the test results and statistical values for various modulation quality parameters. These include:

- Level – the RF power level received by the 4400 (in dBm)

- RMS EVM – the error vector magnitude rms-averaged over a burst. EVM is a relative value and shown as a percentage.

- Peak EVM – the maximum error vector magnitude within a burst. EVM is indicated in per cent.

- 95th% EVM – the error vector magnitude that is not exceeded by 95% of the symbols of a burst. The 95th% EVM is shown in %.

- Frequency Error – the deviation of the carrier frequency compared to the expected carrier. The frequency error is measured in Hertz. The measurement accuracy depends on the time base used (internal TCXO, OCXO or an external reference).

- Origin Offset – The origin offset indicates the accuracy of the I/Q modulator's DC setup; it is usually an undesired leakage produced due to a DC offset in the phone's I/Q modulator. It is measured in dBc.

EGPRS Burst menu The Burst menu allows measurements to determine the power and burst-shaping quality of the mobile under test. Note that the power-time template for 8-PSK modulation differs from that for GMSK. For basic information on burst measurements in synchronous mode, see the GSM or GSM/GPRS chapter of this manual. To access the EGPRS Burst menu, press the Burst marker tab in EGPRS call mode.

Input parameters

Results

The measurement results are displayed in the output fields given below.

Template — Indicates whether the mobile under test satisfies or violates the PTT limits for the multislot operation. The message FAIL is displayed if a measurement in any time slot involved in the test violates the PTT limits. The message PASS is displayed if a measurement in all time slots fulfills the PTT limits.

Level — Display the results of the statistical evaluation of the power level measurements in the time slot set by the input parameter UL-Slot.

EGPRS Burst Zoom menu

To access the EGPRS Burst Zoom menu, press the **Zoom** softkey on the EGPRS Burst menu. Two different menus are accessible on the GPRS Burst Zoom menu via marker tabs:

– To display the power transition phase of the burst, press the RAMP marker tab.
 This menu provides you with specific functions to investigate both the rising and the falling edge of the burst in detail.
– To display the phase error for the active part of the burst, press the Constant Power marker tab.
 This menu enables you to check all details of the part of the burst that contains the information.

The figure below shows the zoom display for the power transition phases of a time slot.
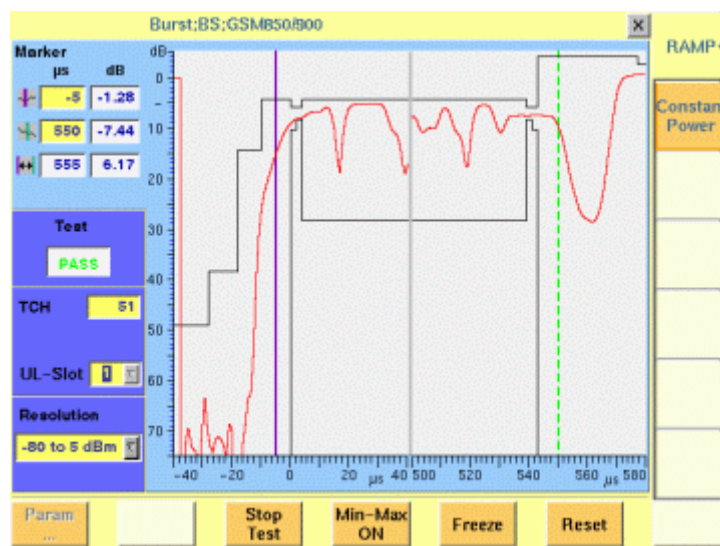
Input parameters

For test parameters to be entered before starting a test, see "Test parameters" on page 161.

Results

The GPRS Burst Zoom menu graphically displays the selected phase of the burst set by the input parameters on the GPRS Burst menu.

The Test output field indicates whether the mobile under test satisfies or violates the power-time template (PTT) limits for the multislot operation. The message FAIL is displayed if a measurement in any time slot involved in the test violates the PTT limits. The message PASS is displayed if measurements in all time slots fulfill the PTT limits.

Display markers

The 4400 offers two markers for easy reading of the graphic display in the burst test menu.

Marker		
	µs	dB
⤓	-5	-2.84
⤒	550	-8.69
↔	555	5.85

On the µs entry fields, you set the markers to the required horizontal positions (= time positions, relative to the beginning of the burst). The corresponding result fields (dB) show the measurement values obtained for the two positions in dB.

The third row shows the difference between the selected time positions and the difference between the two corresponding measurement values.

Test parameters

In this section, the test parameters to be entered before starting a test are summarized for all GPRS menus.

BCCH — Channel number of the base channel in the time slot 0. Valid range:
1...124 for P-GSM 900
0...124, 975...1023 for E-GSM 900
0...124, 955...1023 for R-GSM 900
512...885 for GSM 1800
512...810 for GSM 1900
128...251 for GSM 850
Default is 63.

> **NOTE**
>
> The value of the BCCH carrier cannot be changed during an active UL or DL TBF.

1st slot / No.of slots — Entry field to set which of the sequence of bursts shall be displayed. Valid entries depend on the parameter No. UL-Slots, default is 1/1. The first number designates which of the active time slots the Willtek 4400 shall begin with on the Burst live display.
The second number defines how many from the sequence of bursts shall be displayed.

Marker — Entry fields to set the display markers for reading of the measurement results at specific positions.

No. DL-Slots — Number of PDTCHs to be transmitted by the Willtek 4463 GSM/GPRS System Option and 4468 EDGE System Option. No. DL-Slots successive downlink slots are assigned by the 4463 GSM/GPRS System Option and 4468 EDGE System Option depending on the mobile's capabilities. The value can be entered after the mobile has been GPRS Attached and after it has informed the Willtek 4400 of its multislot class. The value is reset to 1 after a GPRS Detach. Note: The number of downlink slots is selectable only after a GPRS Attach, as the mobile informs the 4400 of its multislot class during the GPRS Attach procedure. The valid range depends on the multislot class n: x/y/z of the mobile under test displayed in the Multislot class field (see "Uplink and downlink assignments" on page 169). No. DL-Slots must be less than or equal to the parameter x, and the sum of No. DL-Slots and No. UL-Slots must not exceed the parameter z.

No. UL-Slots — Number of physical channels to be used by the mobile station. The value is automatically adjusted when a mobile registers and announces a multislot class supporting less channels than defined in this field. No. UL-Slots successive uplink slots are assigned by the 4463 GSM/GPRS System Option and 4468 EDGE System Option depending on the mobile's capabilities.
Note: The number of uplink slots is selectable only after a GPRS Attach, as the mobile informs the 4400 of its multislot class during the GPRS Attach procedure. The valid range depends on the multislot class n: x/y/z of the mobile under test displayed in the Multislot class field. No. UL-Slots must be less than or equal to the parameter y, and the sum of No. DL-Slots and No. UL-Slots must not exceed the parameter z.

TCH — Packet data traffic channel number. Valid range:
1...124 for P-GSM 900
0...124, 975...1023 for E-GSM 900
0...124, 955...1023 for R-GSM 900
512...885 for GSM 1800
512...810 for GSM 1900
128...251 for GSM 850
Default is 63.

NOTE

The value of the TCH carrier can be changed during an active UL and/or DL TBF.

P1, P2, P3, P4 — Four entry fields to set the power levels $Pn = Gamma_0 - Gamma_{CH}$ for the uplink slots, where n is the number of the active time slot within a TDMA frame, for mobile station output power control according to GSM 05.08 (see "Uplink Power Control" on page 170).

NOTE

The power levels can also be changed during an active TBF.

Results for UL-Slot — Number of the slot (burst) for which results are displayed. Valid range is 1...(No. UL-Slots), default is 2.

RF Out — Output level of the 4463 GSM/GPRS System Option and 4468 EDGE System Option in case that all time slots use the same power level. Valid range is –120...–10 dBm.
Note: This entry field is available only if the RF Out Levels Standard check box on GPRS Receiver menus is selected.

EGPRS parameter menus The GPRS Parameter menus allow you to set a number of system parameters including the power/time templates, test limits, the coupling loss curve and the corner points. For basic information on system parameters, see the System Parameters section in the GSM or GSM/GPRS call mode chapter. To access a GPRS Parameter menu, press the **Param...** softkey while working at the test level of any GPRS test.

> **NOTE**
>
> During a test, the GPRS Parameter menus are accessible but the parameters cannot be changed.

There are six different GPRS Parameter menus selectable by the respective marker tabs:

- Basic – see "EGPRS basic parameter menu" on page 163
- PTT – see "EGPRS PTT parameter menu" on page 164
- Limits –see "EGPRS Limits menu" on page 164
- Power Limits – see "EGPRS PTT parameter menu" on page 164
- Coupling Loss – see "Other parameter menus" on page 165
- Corner Points – see "Other parameter menus" on page 165

EGPRS basic parameter menu The EGPRS Basic Parameter menu allows you to set the parameters of the base station simulated by the 4463 GSM/GPRS System Option and 4468 EDGE System Option in EGPRS synchronous test mode.
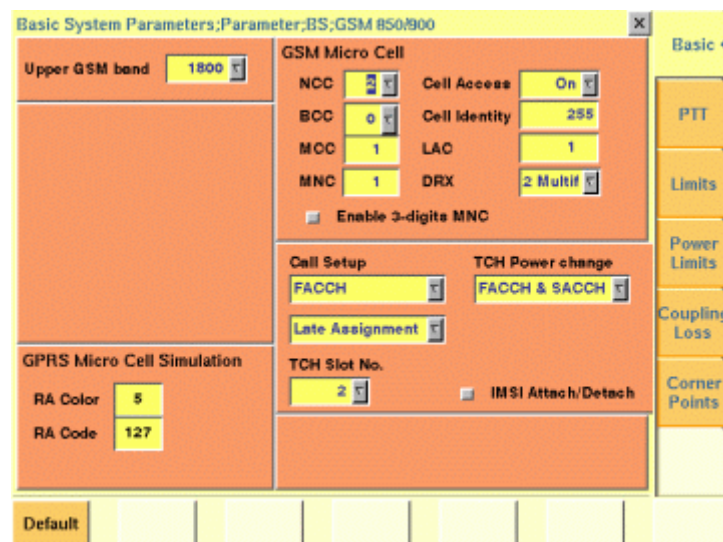
Input parameters

The EGPRS Basic Parameter menu is divided into several areas. All parameter fields except those for the GPRS Micro Cell Simulation are identical to those in the GSM environment; their descriptions can be found in the GSM call mode chapter.

RA Color — Sent in the SI3 and SI4 Rest Octets. The normal use is that different different RA Color codes in adjacent cells indicate to the mobile that the cells belong to different Routing Areas. The valid range is 1…7.

RA Code — The Routing Area Code as broadcast in the SI13 Rest Octets. The valid range is 0…255.

Reset

To reset all parameters in the 4468 EDGE System Option to default values, press the **Default** softkey.

NOTE

Pressing the **Default** softkey will delete all values entered on all GSM/GPRS parameter menus and set them to factory defaults.

Help

To access the Help menu, press the Help key　 ? 　.

EGPRS PTT parameter menu

In EGPRS mode, the power/time template differs from the normal GSM template because the 8-PSK signal does not have a constant envelope. For basic information on power/time templates and how to set the PTT limits, see the GSM or GSM/GPRS chapter.
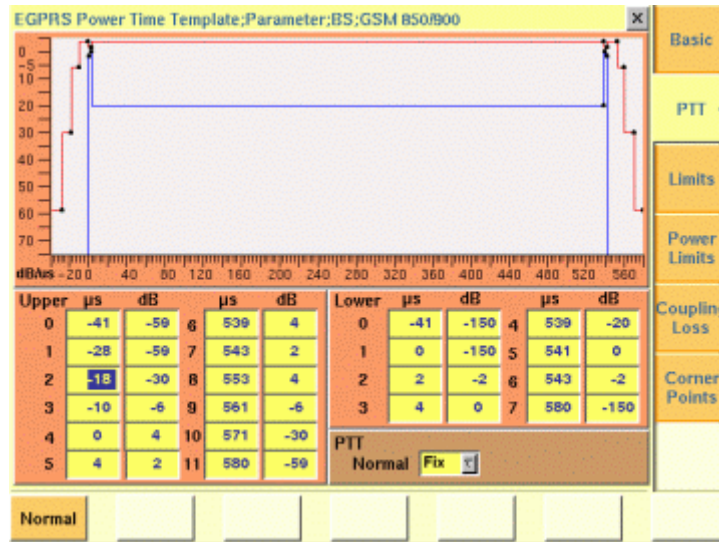
EGPRS Limits menu

In this menu, limit values can be entered for the modulation quality parameters of the Basic and EVM/Frequency menus.

Other parameter menus   The Power Limits, Coupling Loss and Corner Points menus are the same as in GSM and GSM/GPRS.

EGPRS basics

This section covers information on mobile phone testing in EGPRS synchronous mode. Read this section to learn of EGPRS test prerequisites and modes before starting a test using the Willtek 4468 EDGE System Option.

Why EGPRS tests? EGPRS (Enhanced General Packet Radio Service) is an extension of GPRS for higher data rates. These are achieved using a new modulation format: 8-PSK.

EDGE offers a choice between the two modulation formats GMSK and 8-PSK. The latter transmits 3 bits per symbol, in contrast to GMSK modulation where only 1 bit per symbol is transmitted.
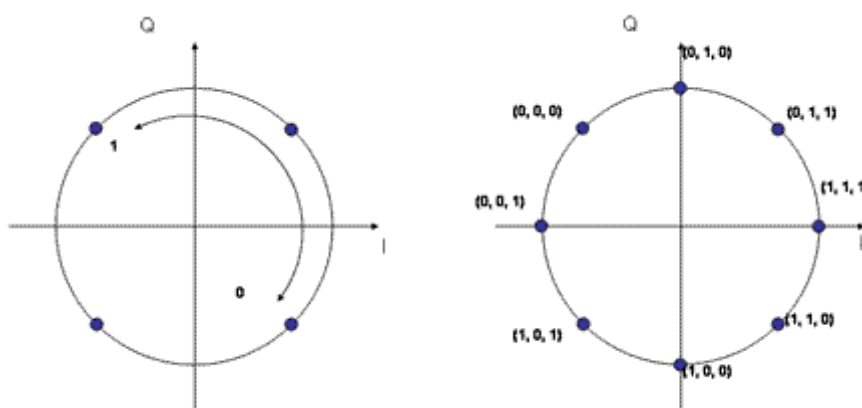
The amplitude of the signals transmitted by the mobile phone is no longer constant because it is not rotating on a circle as it does for GMSK. It can actually change from one phase state to any of the eight phase states, having the effect that the amplitude may cross near the origin of the I/Q plane. In order to avoid this, the whole I/Q plane is rotating by 22.5° per symbol, which ensures that the signal never reaches the origin.

This effect has an impact on the type of amplifier that can be used and may result in separate TX paths for GMSK and 8PSK. Depending on the type of amplifier being used, the battery life will be constrained as well.

Like GPRS, EGPRS uses different coding schemes, combined with either GMSK or 8-PSK modulation. This allows to maintain the connection at reduced data rates under bad radio conditions. The following table shows the different Modulation and Coding Schemes (MCS) for EDGE.

Table 1   Modulation and Coding Schemes

MSC	Modulation	Data rate
MCS–1	GMSK	8.8 kbps
MCS–2	GMSK	11.2 kbps
MCS–3	GMSK	14.8 kbps
MCS–4	GMSK	17.6 kbps

Table 1   Modulation and Coding Schemes

MSC	Modulation	Data rate
MCS-5	8-PSK	22.4 kbps
MCS-6	8-PSK	29.6 kbps
MCS-7	8-PSK	44.8 kbps
MCS-8	8-PSK	54.4 kbps
MCS-9	8-PSK	59.2 kbps

The new modulation requires a completely new transmitter design. Due to the fact that the envelope is no longer constant and therefore the amplitude is changing for EDGE, the power amplifiers used are different. For this reason a common approach for RF designers is to split the transmit path into a path for GMSK transmission and one for 8-PSK transmission. The 8-PSK modulator and amplifier should be tested and in some cases should be aligned separately to ensure that they work to specifications.

The physical receive path can remain the same for GMSK and 8-PSK. There is no need for having a second receive path. This means that the existing receiver tests for GMSK (GSM or GSM/GPRS) are sufficient and do not need to be extended.

Required tests   As explained above, only the transmitter needs testing and alignment. The tests can be divided into power and modulation quality tests.

Power measurements   One important quality factor with all RF systems is transmit power. The maximum power level is key to a good data transfer rate in rural areas with a large cell radius. As the transmitter is in a different operating mode when applying 8-PSK modulation, measurements are essential to ensure that the power level lies within the allowable limits. Note that the cell radius can be lower with 8-PSK modulation than with GMSK due to the increased number of symbols.

To minimize unnecessary interference with adjacent cells and to ensure proper handover between cells, the power control mechanism in GSM and GPRS uses further power steps with exact levels. In manufacturing and after repair, these levels are usually calibrated with the help of a test set, with correction values being stored in the mobile phone.

In measurements of power versus time, the same principles as in GMSK are applied to 8-PSK, but with a modified template; the template takes into account that the 8-PSK signal amplitude varies between symbols, depending on the exact symbol sequence. This attribute of 8-PSK makes it more difficult to use cheap, nonlinear amplifiers and may lead to hardware and/or software designs correcting these nonlinearities. The power/time template for 8-PSK signals is shown below; the flat part at the beginning and the end of the burst applies to the tail bits, which have been chosen such that only minimal variations of the power level occur. The limits of the power level steps are identical to those known from GMSK modulation.
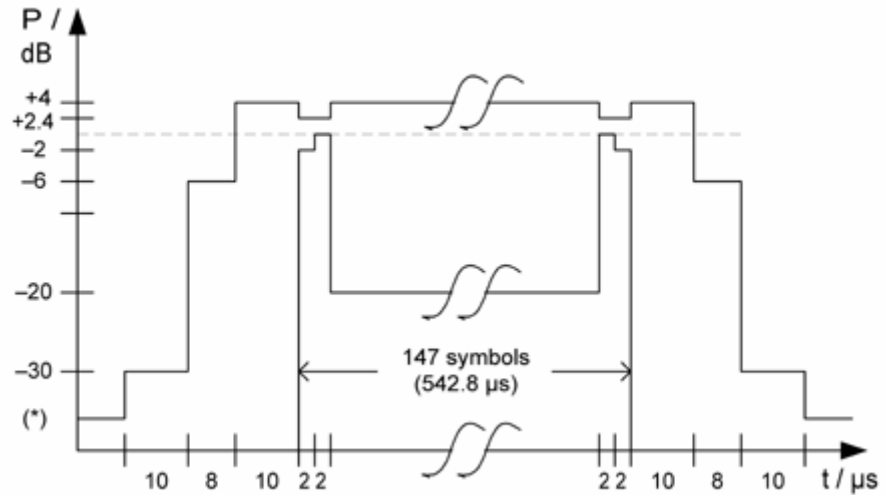
Modulation quality   8-PSK requires modulation measurements different from GMSK. The beauty of these measurements is that they provide parameters which are more likely to allow you to track down the source of any potential.

The key measurement is that of the error vector magnitude or EVM for short, which is the distance in the I/Q diagram between the measured signal and the ideal one (see figure below); the EVM is measured separately for each symbol. Standard measurement results to observe are the RMS–averaged EVM (for all the symbols of a burst), the maximum EVM within the burst, and the 95% EVM. The latter is the error vector magnitude not exceeded by 95% of the symbols within a burst; the parameter effectively disregards the highest 5% of all error vectors.
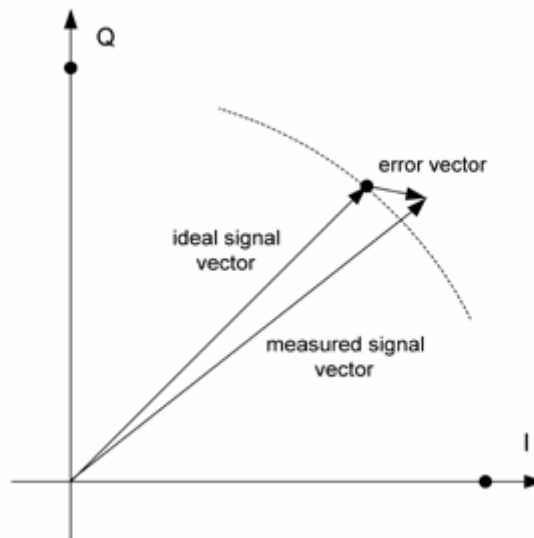
The error vector can be broken down into two components; one component is radial i.e. in the direction of the circle (phase error) and the other is vertical to the circle (magnitude error). These parameters are not part of the mandatory measurements but can be helpful in tracing a problem with the modulator.

Another parameter which is determined inside the test set but not necessarily shown or assessed is the amplitude change or droop, a measure of the stability of the power level during the burst. An amplitude change may be caused by on-chip temperature changes during the RF transmission.

Finally, if the I and/or the Q component of the modulator has an unwanted DC component, this can be determined in the origin offset suppression measurement.

GPRS EDGE uses an extended version of the GPRS (General Packet Radio Service) protocol. GPRS replaces the GSM protocol above the physical layer; GSM and GPRS operation share the same physical resources. In manufacturing and service, the purpose of testing is to ensure that the hardware is working correctly, rather than testing the protocol software.

Some GSM/GPRS/EGPRS mobile phones may have single-slot capability only. Although the RF hardware is the same for GSM and GPRS and is usually tested in circuit-switched mode tests, there may be an additional 8-PSK transmitter for EDGE which should be tested.

Many EGPRS mobile phones support multiple time slots in the downlink and uplink. This capability challenges the mobile phone more than in single-slot operation: Components may produce side effects due to voltage drops or additional heat when multiple slots are transmitted or received. Therefore, most manufacturers ensure product quality by multislot tests.

Uplink and downlink assignments GPRS is used to transmit packets or blocks of data from one end to another. While circuit-switched connections are bidirectional and block a channel in both directions for the duration of a call, packet data transfers may be unidirectional and occupy the channel only for the time of the block transfer, i.e. the resource "channel" can be shared by many users and allocated to users on a block-by-block basis.

The channel and block administration is performed by the base station. When one party needs to transfer data, a so-called Temporary Block Flow (TBF) is initiated on RLC/MAC level. Since the information flow is unidirectional, there is a separate TBF for each direction, called Downlink TBF and Uplink TBF. Moreover, each TBF can use one or multiple time slots. A bidirectional data transfer can be asymmetric, i.e. may use more time slots in one than in the other direction.

The GPRS protocol uses a number of new protocol layers and procedures. For the purpose of RF testing, only RLC and MAC layers are required, plus a few procedures such as the GPRS Attach (registration procedure of a mobile in a GPRS network). Higher layers (e.g. LLC) and their procedures, such as PDP Context Activation, are not required for RF testing and therefore not supported by the 4463 GSM/GPRS System Option and 4468 EDGE System Option.

<table>
<tr><td>Assignment of time slots for
uplink and downlink</td><td>A GPRS mobile phone is associated to a multislot class; each multislot class determines how many time slots the mobile can transmit and/or receive, and also defines the minimum duty cycle between transmit and receive slots. The most important parameters are:</td></tr>
</table>

Assignment of time slots for uplink and downlink

A GPRS mobile phone is associated to a multislot class; each multislot class determines how many time slots the mobile can transmit and/or receive, and also defines the minimum duty cycle between transmit and receive slots. The most important parameters are:

– The maximum number of downlink slots (without any assigned channel in the uplink)

– The maximum number of uplink slots (without any assigned channel in the downlink)

– The maximum number of combined slots assigned in downlink and uplink at any time

Example: A mobile of multislot class 10 can use up to four time slots in the downlink and up to two time slots in the uplink, but only up to five time slots in total (e.g. 4 DL + 1 UL or 3 DL + 2 UL).

After a GPRS Attach, the multislot class is shown on the GPRS Basic menu together with these capabilities; in the above example, it would display "10: 4/2/5" meaning that the mobile is of multislot class, supporting 4 DL, 2 UL and 5 time slots in total.

NOTE

Changing the number of downlink or uplink slots is supported only when the mobile is GPRS attached. The Willtek 4400 only allows you to enter permissible combinations of downlink and uplink slots, depending on the multislot class.

Uplink Power Control

In a real network, the transmit power of the mobile phone can be controlled by the network directly (close loop power control) and, depending on the setting of the open loop power control parameter Alpha, partially also by the mobile phone (open loop power control). In case of the open loop power control, the transmit power depends on the received signal strength.

A real network can even set different uplink power levels for the individual time slots assigned to the mobile. This is also supported by the 4400; the power levels for the individual uplink time slots can be defined on the EGPRS Burst menu.

Test prerequisites

In order to be able to make tests in synchronous mode (i.e. with proper channel assignments), the mobile phone must be GPRS attached. The GPRS Attach is a procedure which usually informs the network of the mobile's presence and GPRS capabilities within a GPRS cell. The sequence of events during a GPRS Attach is as follows:

1 When the GSM/GPRS synchronous mode menus are active, the 4400 simulates a GSM network with GPRS capabilities, i.e. it broadcasts in one of the System Information Messages (type 13) that it supports GPRS data transfer. The GPRS Attach field in the call graph of the GPRS Basic menu is shown with white background.

2 When the mobile is switched on, it scans the frequency bands for a valid network. When it finds and accepts the 4463 GSM/GPRS System Option and 4468 EDGE System Option GSM/GPRS network, it may (or may not) perform a Location Update for its GSM presence in the network.

3 The mobile phone will have to initiate a GPRS Attach procedure. Depending on the implementation in the mobile phone, this could be done in one of two ways:

– Some mobile phones initiate the GPRS Attach automatically after they selected a network which has GPRS capabilities.

– In other mobile phones, the GPRS Attach is initiated only when the user requests a GPRS service, e.g. by selecting an appropriate entry on the user interface of the mobile phone.

4 The 4400 and the mobile will communicate via UL/DL TBFs to exchange protocol information. When the procedure is concluded successfully:

– The mobile phone will usually indicate that it is registered to a GPRS network (e.g. by the word "GPRS" in its display).

– The 4400 will indicate the multislot class of the mobile in the Multislot class field on the EGPRS Basic menu.

– The background of the EGPRS Attach field in the call graph of the EGPRS Basic menu will turn green.

– No TBF will be active.

GPRS test mode The 4468 EDGE System Option uses the GPRS Test Mode and applies basically the same protocol as in a GPRS BLER-USF test.

EDGE modulation and coding schemes The 4400 uses MCS-1 for signaling and assigns and packet data traffic channel with MCS-5 for uplink user data transfer. MCS-5 applies 8-PSK modulation.

EDGE Asynchronous Mode Operation

<div style="text-align: right; border: 1px solid black; display: inline-block;">5</div>

This chapter provides task-based instructions for using the asynchronous mode (Generator/Analyzer) of the **4468 EDGE System Option**. Topics discussed in this chapter are as follows:

Overview

The generator/analyzer mode of the 4468 EDGE System Option provides test and measurement tools for asynchronous measurements; these are measurements without call setup and without signaling.

This section consists of the following parts:

– RF generator – This section describes the menu of the RF signal generator. Using the RF generator, continuous or burst signals can be generated, either unmodulated, with GMSK modulation or with AM.

– EVM/frequency analyzer – The 4400 analyzes the modulation quality of the received bursts.

– Burst analyzer – Similar to the standard burst measurement, the 4400 will analyze the shape of the bursts received using the EDGE power/time template.

– Alignment – A predefined series of checks and alignments can be carried out, depending on the type of mobile phone.

– Parameter menus – The most frequently used parameters can be adjusted to your needs in the respective generator or analyzer menu. Some additional parameters are available that affect the generated signal and the measurement views; these parameters can be modified in the parameter menus.

To gain access to the menus of the generator/analyzer tool:

1 Go to the Welcome menu (see picture below).

2 Push the marker tab for the corresponding system option: **EGPRS**.

3 Push the **Gen/Ana** softkey.

RF generator

To gain access to this menu, push the [RF Gen] marker tab while working with the generator/analyzer tool.

This menu allows to set the parameters of the RF Generator.

The RF generator can be set to generate radio signals from 800 to 2000 MHz; either as continuous or as burst signals.

Modulation can be GMSK – or the signals can be totally unmodulated in order to stimulate pure RF circuitry.

The RF generator menu consists of the following main areas:

– The **system area** allows to set the generator frequency either in MHz or in GSM channel numbers.

– The **signal type area** is where you select bursted or continuous signals.

– The **modulation area** allows to set the modulation parameters.

– The **RF Out area** allows to define individual output levels for the different time slots when a multislot signal type is selected.

Parameters of the system area

The output frequency of the RF generator and the RF output power level are set in this area.

System — Selection field to set the frequency range of the channel numbers above 511.

Selections possible: GSM 900 / 1800 or GSM 900 / 1900.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

For details regarding GSM frequency ranges, check with section "Basic specifications of GSM bands" on page 97.

uplink / downlink — Selection field to specify either the uplink or the downlink frequency range.
Selections possible: uplink or downlink.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

For details regarding GSM frequency ranges, check with section "Basic specifications of GSM bands" on page 97.

Freq — Entry field to set the output frequency of the RF generator.
Entry range: 800 to 1000 MHz and 1700 to 2000 MHz in increments of 10 Hz.

NOTE

When the frequency entered corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display the channel number and an offset (if any) on the two entry fields of this line.

In case the optional frequency extension has been installed on your 4400, the frequency range available will be different (see specifications of the frequency extension option for details).

Channel — Entry field for the GSM channel number.
Entry range: 0 to 1023

NOTE

When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display the corresponding channel number as the default on this entry field.

Offset — Entry field for the offset from the GSM channel frequency specified.
Entry range: –99.99 to 99.99 kHz in increments of 10 Hz.

NOTE

When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display any offset from the nominal GSM channel frequency as the default on this entry field.

Level — Entry field for the RF output power level of the RF generator. The output signal will be available on the RF in/out connector.
Entry range: –120.0 to –10.0 dBm in increments of 0.1 dB.

> **NOTE**
>
> The level set on this entry field will only be correct when the load impedance is 50 ohms.

> **NOTE**
>
> This entry field is not accessible if individual power levels are set, i.e. if the Signal type is set to multislot and the check box RF Out Standard is selected.

ON / OFF

Display field for the current output status of the RF generator. When ON is displayed, RF signals will be output; when OFF is displayed, the RF generator will be silent.
To toggle between the two states, use the **Gen. Off** respectively the **Gen. On** softkey.

Parameters of the signal type area

The type of the RF signal to be generated is set in this area of the menu.

Signal type — Selection field.
Selects one of the following signal types: continuous (with a constant power level), burst (single-slot) or multislot. The latter can have individual power levels at each of the eight time slots per TDMA frame.

> **NOTE**
>
> Setting "continuous" means that there will be a constant RF output power level (without power ramping). The generated RF signal can still be GMSK modulated (see description of the menu area below).

> **NOTE**
>
> When "continuous" has been selected on this field, the Timeslot entry field will not be available.

> **NOTE**
>
> In case "continuous" has been selected together with GMSK modulation, all time slots will be filled with data and a training sequence (if selected).

Timeslot — Selection field to specify the time slot during which the burst will be generated.
Selections possible: 0 to 6.

---- **NOTE**

When "continuous" has been selected on the Signal type entry field, this entry field will not be available.

---- **NOTE**

The internal frame synchronization signal of the 4400 will be available on the **SYNC IN/OUT** connector on the rear of the test set.

Parameters of the modulation area

The modulation of the RF signal to be generated is set in this area of the menu.

Modulation — Selection field.
Selections possible: GMSK or none.

---- **NOTE**

Setting "none" means that the RF output signal generated by the 4400 will be unmodulated, i.e. a pure carrier sine wave.

---- **NOTE**

When "none" has been selected on this field, all other entry fields of this menu area will not be available.

---- **NOTE**

The modulation setting also affects measurements in the analyzer: If GMSK is selected, the analyzer will attempt to demodulate the signal; phase and frequency error are shown with respect to a GMSK-modulated signal. Adversely, if no modulation is selected in the generator, the analyzer shows the results assuming that the input signal is unmodulated as well.

Channel — Selects one of the following channel types: GSM, PDCH or BCH+PDCH.
For GSM channel, the 4463 GSM/GPRS System Option and 4468 EDGE System Option generates a modulated signal with a data pattern specified in the Data entry field. If the check box Training Seq. is selected, the data pattern is filled into two blocks of 58 "encrypted bits"; otherwise, the data pattern is filled into all 148 bits of the burst. No channel coding is performed.
Depending on the signal type selected in the Signal type entry field, the signal is sent either continuously (with constant power level or individual power levels) or as bursts. It does not take into account any tail bits or multiframe structures with idle frames.
If PDCH is selected, a signal with one or multiple time slots of channel combination 13 (PDTCH/F + PACCH/F PTCCH/F) is generated. The PDTCH/F blocks can be split into RLC/MAC headers with user-definable contents and a pseudorandom

bit sequence PRBS (selectable in the Data entry field) as data payload. If multislot signal is selected in the Signal type entry field, the time slots of each TDMA frame are filled with exactly the same contents.

> **NOTE**
>
> This entry field is available only for burst or multislot signal type.
> The BCH+PDCH channel type is available for the multislot signal type only. At the time slot 0, the 4463 GSM/GPRS System Option and 4468 EDGE System Option transmits a physical channel with channel combination 5 (FCCH + SCH + BCCH + CCCH + SDCCH/4 + SACCH/4). The BCCH includes transmission of System Information Message Type 13, which relays information about GPRS capabilities of the simulated network. The bit pattern selectable in the Data entry field does not apply to the time slot 0. The other time slots (1...7) are filled with identical PDCH data, see description of PDCH above.

> **NOTE**
>
> Some of the parameters transmitted in the System Information Messages on the BCCH can be affected on the Basic Parameter menu.

CS — Selects the coding sequence in the range 1...4 if PDCH or BCH+PDCH is selected in the Channel entry field. This parameter refers to the coding algorithm on the PDTCH.

Data — Selects the data payload for GSM channels (uncoded) and packet data traffic channels.
For GSM channel selected in the Channel entry field, valid entries are 0...0, 1...1, 1010..., 1100..., 11110000, 1...10...0, PRBS-9, PRBS-15, PRBS-23.
If PDCH or BCH+PDCH is selected in the Channel entry field, valid entries are PRBS-9, PRBS-15, PRBS-23.

Training Seq. — Check box and selection field.
With the check box, transmission of a training sequence in each time slot can either be selected or deselected.
If transmission of a training sequence has been selected, the training sequence code can be chosen (range of the selection field: from 0 to 7).

> **NOTE**
>
> Each one of the eight training sequence codes refers to a predefined 26-bit training sequence for normal bursts.

> **NOTE**
>
> When a training sequence has been selected, it will be transmitted in the middle of every time slot. Thus, there will be some kind of time slot structure of the RF signal generated, even when "continuous" has been selected as the Signal type.

> **NOTE**
>
> However, there will be no GSM frame structure as the RF Generator will not provide a BCCH (and thus, there will be no mark for a slot '0' apart from the internal frame synchronization signal of the 4400, which is available on the **SYNC IN/OUT** connector on the rear of the test set).

RF Out Standard — If selected, the generator will apply the RF output level entered in the Level field to all time slots, and the RF Out Timeslot entry field for individual time slots will not be accessible.

NOTE

This check box is available only for the multislot signal type.

RF Out Timeslot — Allows setting the RF power level of each time slot (0...7) if the RF Out Standard check box is not selected. For each time slot, any RF power level within the specified range (usually –120...–10 dBm) is permissible.
The following softkeys are available on the Generator menu:

– The **Gen On/Gen Off** softkey turns the generator on and off.
After pressing the **Gen On** softkey, the generator transmits the signal configured, and the softkey label turns to **Gen Off**. Pressing the **Gen Off** softkey turns the generator off and changes the softkey label to **Gen On**.

– The **Param...** softkey gives access to Parameter menus

EVM/frequency analyzer

To gain access to this menu, push the [Analyzer Phase/Freq.] marker tab while working with the generator/analyzer tool.
This menu displays the results of the modulation and frequency error measurements of the RF signal received at the **RF IN/OUT** connector.

NOTE

The analyzer will measure the signal and display the results assuming that the signal is 8-PSK-modulated if the 4400 generator is set to GMSK. If the generator of the 4400 is set to transmit an unmodulated signal, the measurements in the analyzer are calculated assuming that the signal at the RF input connector is unmodulated as well.
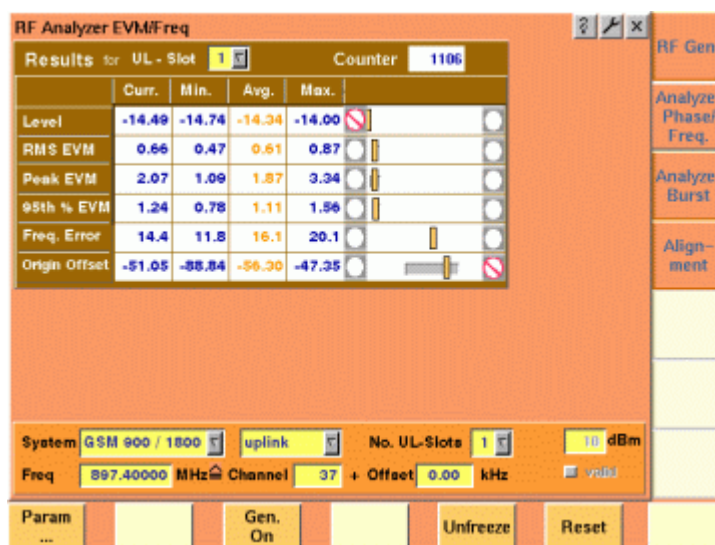
Using the EVM/frequency analyzer

The RF analyzer takes measurements in asynchronous mode, without any call processing functions.

Any Coupling loss table defined in the appropriate Parameters menu will be used for asynchronous measurements.

Before starting a measurement, the radio frequency to be measured on has to be entered.

There are entry fields available for these parameters in the lower area of the menu, just above the softkeys.

The measurement results, resolution settings, and display markers are the same as the corresponding areas and fields of the call mode test menu.

Parameter fields of the RF analyzer

The radio frequency to measure on and the trigger functions are set in this menu area.

System — Selection field to set the frequency range for the channel numbers above 511.

Selections possible: GSM 900 / 1800 or GSM 900 / 1900.

> **NOTE**
>
> To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

> **NOTE**
>
> For details regarding GSM frequency ranges, check with section "Basic specifications of GSM bands" on page 97.

uplink / downlink — Selection field to specify either the uplink or the downlink frequency range.

Selections possible: uplink or downlink.

> **NOTE**
>
> To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

> **NOTE**
>
> For details regarding GSM frequency ranges, check with section "Basic specifications of GSM bands" on page 97.

No. UL–Slots — Selects the expected number of uplink time slots in the range 1…4.

Freq — Entry field to set the measurement frequency for the RF analyzer.
Entry range: 800 to 1000 MHz and 1700 to 2000 MHz in increments of 10 Hz.

> ───── **NOTE**
>
> When the frequency entered corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display the channel number and an offset (if any) on the two entry fields of this line.

> ───── **NOTE**
>
> In case the optional frequency extension has been installed on your 4400, the frequency range available will be different (see specifications of the frequency extension option for details).

Channel — Entry field for the GSM channel number.
Entry range: 0 to 1023

> ───── **NOTE**
>
> When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display the corresponding channel number as the default on this entry field.

Offset — Entry field for the offset from the GSM channel frequency specified.
Entry range: –99.99 to 99.99 kHz in increments of 10 Hz.

> ───── **NOTE**
>
> When the frequency entered on the Freq entry field corresponds to a valid GSM channel number (on the basis of the current settings of the two selection fields above this entry field), the 4400 will automatically display any offset from the nominal GSM channel frequency as the default on this entry field.

Softkeys of the RF Analyzer

Param ... This softkey leads you to the parameter screens where you can set up coupling loss factors and limits.

Gen. Off A push on this softkey switches the RF generator either on or off.

Freeze To freeze the current display for further or detailed analysis (or for printing), push this softkey.
The softkey will change to **Unfreeze**.
Note: After the **Freeze** softkey has been pushed, all measurements will continue in the background, but the display will not be updated.

Reset A push on this softkey resets the statistic evaluation of the measurement results. The measurement counter will be reset as well.

Result fields   For basic information on results provided on the EVM/Frequency menu, see "Required tests" on page 167. The results displayed on the EVM/Frequency menu apply to the slot set by the input parameter Results for UL Slot.

The EVM/Frequency display shows the test results and statistical values for various modulation quality parameters. The parameters are:

– Level – the RF power level received by the 4400 (in dBm)

– RMS EVM – the error vector magnitude rms-averaged over a burst. EVM is a relative value and shown as a percentage.

– Peak EVM – the maximum error vector magnitude within a burst. EVM is indicated in per cent.

– 95th% EVM – the error vector magnitude that is not exceeded by 95% of the symbols of a burst. The 95th% EVM is shown in %.

– Frequency Error – the deviation of the carrier frequency compared to the expected carrier. The frequency error is measured in Hertz. The measurement accuracy depends on the time base used (internal TCXO, OCXO or an external reference).

– Origin Offset – The origin offset indicates the accuracy of the I/Q modulator's DC setup; it is usually an undesired leakage produced due to a DC offset in the phone's I/Q modulator. It is measured in dBc.

Burst analyzer

To gain access to this menu, push the [Analyzer Burst] marker tab while working with the generator/analyzer tool.
This menu displays the results of the asynchronous burst measurements of the RF signal received at the RF in/out connector.

NOTE

The analyzer will measure the signal and display the results assuming that the signal is 8-PSK-modulated if the 4400 generator is set to GMSK. If the generator of the 4400 is set to transmit an unmodulated signal, the measurements in the analyzer are calculated assuming that the signal at the RF input connector is unmodulated as well.

Using the burst analyzer

The RF analyzer takes measurements in asynchronous mode, without any call processing functions.

Any Coupling loss table defined in the appropriate menu will be used for asynchronous measurements as well.

Before starting a measurement, the radio frequency to be measured on has to be entered.

There are entry fields available for these parameters in the lower area of the menu, just above the softkeys.

The measurement results and the display markers are similar to the corresponding areas and fields of the call mode test menu for EDGE.

The graphic display shows the measured RF power versus time, relative to the average power level in the displayed time slot with the highest power. The graph of the power level as a function of time can be shown for one or more time slots. From the up to four time slots that can be received by the 4400, any combination of consecutive bursts can be measured using the First Burst / Number of Bursts entry field.

Parameter fields of the RF analyzer

The radio frequency to measure on and the trigger functions are set in this menu area.

System — Selection field to set the frequency range of the channel numbers above 511.

Selections possible: GSM 900 / 1800 or GSM 900 / 1900.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on the Channel entry field or enter a frequency that corresponds to a valid GSM channel number on the Freq entry field.

NOTE

For details regarding GSM frequency ranges, check with section "Basic speci-
fications of GSM bands" on page 97.

uplink / downlink — Selection field to specify either the uplink or the downlink
frequency range.
Selections possible: uplink or downlink.

NOTE

To gain access to this entry field, either enter a valid GSM channel number on
the Channel entry field or enter a frequency that corresponds to a valid GSM
channel number on the Freq entry field.

NOTE

For details regarding GSM frequency ranges, check with section "Basic speci-
fications of GSM bands" on page 97.

Freq — Entry field to set the measurement frequency for the RF analyzer.
Entry range: 800 to 1000 MHz and 1700 to 2000 MHz in increments of 10 Hz.

NOTE

When the frequency entered corresponds to a valid GSM channel number (on
the basis of the current settings of the two selection fields above this entry
field), the 4400 will automatically display the channel number and an offset
(if any) on the two entry fields of this line.

NOTE

Depending on the installed RF module, the frequency range and power level
range may extend the values specified above.

Channel — Entry field for the GSM channel number.
Entry range: 0 to 1023

NOTE

When the frequency entered on the Freq entry field corresponds to a valid
GSM channel number (on the basis of the current settings of the two selec-
tion fields above this entry field), the 4400 will automatically display the cor-
responding channel number as the default on this entry field.

Offset — Entry field for the offset from the GSM channel frequency specified.
Entry range: –99.99 kHz to 99.99 kHz in increments of 0.01 kHz.

NOTE

When the frequency entered on the Freq entry field corresponds to a valid
GSM channel number (on the basis of the current settings on the two selec-
tion fields above this entry field), the 4400 will automatically display any off-
set from the nominal GSM channel frequency as the default on default on this
entry field.

No. UL–Slots — Selects the expected number of uplink time slots in the range 1...4.

Result for UL–Slot — Selects an uplink time slot within 1...4 to display numerical results.

First Burst / Number of Bursts — The first number and the second number in this entry field indicate the first burst displayed and the total number of bursts displayed respectively.

Softkeys of the RF Analyzer

Param ...	This softkey leads you to the parameter screens where you can set up coupling loss factors and limits.
Gen. Off	A push on this softkey switches the RF generator either on or off.
Freeze	To freeze the current display for further or detailed analysis (or for printing), push this softkey. The softkey will change to **Unfreeze**. **Note:** After the **Freeze** softkey has been pushed, all measurements will continue in the background, but the display will not be updated.
Reset	A push on this softkey resets the statistic evaluation of the measurement results. The measurement counter will be reset as well.

Alignment

The Alignment screen allows you to interactively run a sequence of checks and alignments on the mobile phone in asynchronous mode. An individual sequence can be defined for each type of mobile phone; the phone type is easily selectable. For a selected phone, each step can be carried out on user request, either in the given sequence or out of sequence. Each step can consist of a continuous or burst signal being sent to the phone for stimulation, and/or a signal from the phone being measured in terms of power or spectrum.

This section is divided into two subsections. The first one shows you how to use the Alignment screen; so this is what you need to know when testing a phone. The second one explains you how to set up your own mobile phone scripts. This subsection is applicable to those who want to write a new script, for example for a new type of mobile phone.

Application Many mobile phones are checked, repaired and aligned according to the manufacturer's test procedures. These procedures consist of test steps typically involving stimulation of the mobile phone with a modulated or unmodulated signal. In most cases, the RF power or the modulation spectrum is measured at

a test point, and compared with given limits. For modulator alignment purposes, the RF power at 67 kHz is often compared with the power at the nominal carrier frequency.

For these purposes, the 4400 provides a screen combining signal generator, power and spectrum analyzer functions with the scripting functionality to give a powerful tool for quick and precise tests and alignments.

The Alignment screen allows you to select a script for a mobile phone. In each script, a number of test steps are defined – just as in the service manual for the respective phone. These test steps can be executed in sequence; alternatively, they can also be performed individually as required.

Each test step can be selected by name; ideally, this name is identical with the name of the test step in the service manual.

Alignment screen elements

As you can see in the picture above, the Alignment screen is divided into three sections:

– The upper half of the screen shows the RF spectrum. Results are only shown or updated when required – in some measurement steps, you will only need the RF power reading.

– The spectrum display contains three markers; the marker section in the lower left part of the screen allows you to adjust their respective horizontal positions and to read the results.

– The lower right section allows you to select the type of mobile phone to be tested and to choose a test step. It also displays the most important RF settings and shows the measured RF power.

Selecting a mobile phone and test step

Two scroll fields allow you to select the type of mobile phone under test and the next test step for the selected phone, respectively.

Type — This scroll field allows you to select a mobile phone. The 4400 is delivered with a few examples; scripts for more mobile phones may be available from the mobile phone manufacturers, from Willtek's support team or from Willtek's web pages.

Step — For each selectable mobile phone, a number of test steps are available. A new step can be selected by its name from the Step scroll box. In each new step, generator and analyzer settings may change as well as the type of measurement. The most important generator (Gen) and analyzer (Ana) settings are shown below the Step scroll box.
Instead of selecting a step with the scroll box, you can select a new step with the softkeys (see below).

Results and Markers   The **Spectrum** display can show you the spectrum within 100 kHz around the center frequency. The result is displayed relative to the peak; this means that the highest point in the frequency spectrum is always 0 dB.
The center channel and frequency are indicated below the selection fields for phone type and test step (Ana TCH, Freq). The spectrum is measured in a 10 kHz resolution bandwidth.

Marker kHz		dBm
▽	-100.0	---
▼	0.0	---
▼	100.0	---
↔	100.0	---
↔	200.0	---
↔	100.0	---

Three markers are available, allowing you to read the numerical results as absolute values and relative to each other.

Many tests require a reading of the **RF power**. The RF power (burst peak power) is shown on the right-hand side, measured in dBm.

A measurement counter indicates how many measurements have been performed and displayed already since the start of the test. This can be useful in order to verify that measurements are still being performed. It can also be used to monitor the progress when averaging results; the averaging functions can be selected by softkey (see below).
The counter and all measurement results are reset with the Reset softkey.

Softkeys on the alignment screen   The softkeys provide different functions to control the functions and the result output:

Start Avg.   This softkey can be used to start an averaging function over the measurements. It affects both the spectrum measurements and the RF power measurements.

Previous Step

If you want to repeat a test that is one step above the current test (in the list of test steps), a push on this softkey is sufficient to stop the current test and start the wanted test step.

The list is circular: If there is no previous step available because the present step is the first one in the test list, the 4400 continues with the (last) entry at the end of the list.

Repeat Step

This softkey can be used to restart the test step displayed in the Step field. This may be useful after halting a test, moving to another test menu and returning to this menu.

Next Step

For tests in the defined sequence, this softkey serves to stop the current test and start the next one.

The list is circular: If there is no test step following the current one because the end of the test list has been reached, the 4400 continues with the first entry in the test list. This could be used to start a new test with another phone when the previous phone is done.

Freeze

A push on this softkey stops the measurement update, keeping the last result on the display.

The softkey description changes to Unfreeze, allowing you to continue the measurement update after another push on this button.

Reset

The Reset button restarts the ongoing measurements and sets back the measurement counter. If averaging is active, previous results are discarded and the averaging process is restarted from the next new measurement result.

Writing your own alignment script

In each step, the generator and analyzer settings may be changed and a measurement is usually started. All the steps for one type of mobile phone are defined in a script file. This script is run once to initialize the tests when the type of phone is selected, and later on each time a new test step is selected. The script is stored in ASCII format on the hard disk of the 4400; the location is /rapid/param. The file name should follow the "filename".BAS format, where "filename" consists of at least one character. Ideally, the file name represents the name or designation of the mobile phone.

To create a new script file or modify an existing one, you can either use a standard editor on an external PC or the built-in RAPID! programming environment which includes an editor.

Existing files can be loaded from floppy disk onto the 4400's hard disk. Please ensure that the file is stored in the correct directory!

Alignment scripts basically follow the language rules and conventions as simple RAPID! programs. See section "RAPID! syntax" on page 279 about RAPID!. Some additional rules apply:

- **List of test steps:** The preparation includes the definition of the names of the steps which should appear in the Step scroll box. This definition is given by the `gui "STEP:a,b,c,...,"` statement. a,b,c,… are to be replaced by the text for the different steps (cases).
  If you want to change the sequence of steps, just reorder them in the `gui` statement.

- **Initialization:** An additional initialization phase sets the 4400 to a known state, for example with settings that are common for all the test steps such as the coupling loss. All these preparatory commands, including the list of test steps, are placed in an empty case statement.

  Example:
  ```
  case "" ' Elements of the scroll box
  gui "STEP:Ch 56 -55dBm,Ch 700 -85dBm,Ch 700 TX-IQ"
  scpi ":RFGenerator:STYPe CWAVe"
  scpi ":CONFigure:COUPloss:DATA ""3210_1.5_2"", 897.6,
  1.5, 1747.8, 4.0"
  scpi ":CONFigure:COUPloss:STATe ON"
  ```

- **Definition of the actual step:** The actual step is a case. The case is handed over to the script in the following statement: `select case cmdline$`. This statement must precede the first case.

- **Test steps:** Each test step is located in a `case` statement. The text in the case statement is the one presented in the Step scroll box. See example below.

- **User instructions** before the start of a measurement should be placed in a message box. Example:
  ```
  msgbox "Settings done","OK"
  status = inkeywait
  ```

- **Progress indication text:** You can display a message, for example to explain the ongoing test step. Example:
  ```
  open "GUI:" as #1
  sub gui(cmd$)
      print #1, cmd$
  end sub
  gui "STEP:Ch 60, -55dBm"
  ```

- **Setup and measurement commands:** You can issue 4400 SCPI commands by sending them to a device called "SCPI:". Example:
  ```
  open "SCPI:" as #2
  sub scpi(cmd$)
      print #2, cmd$
  end sub
  scpi ":RFGenerator:STYPe CWAVe"
  ```

Example:

```
rem -------- BigTel 1234 ----------
open "GUI:" as #1
open "SCPI:" as #2

sub gui(cmd$)
print #1, cmd$
end sub
```

```
sub scpi(cmd$)
print #2, cmd$
end sub

rem "This is an automatic sequence to align the BigTel
1234"

select case cmdline$
case "" ' Setting up the scroll box
gui "STEP:Ch 60 -80dBm,Ch 700 -50dBm,Ch 700 TX-IQ,"
scpi ":RFGenerator:STYPe CWAVe"
scpi ":CONFigure:COUPloss:DATA
""BigTel_1234_with_RF_cable"", 902.0, 1.5, 1742.8,
2.0"
scpi ":CONFigure:COUPloss:STATe ON"

case "Ch 60 -80dBm"
scpi ":RFGenerator:GSM:STATe ON"
scpi ":RFG:GSM:MODulation:TSEQuence:STATe OFF"
scpi ":RFGenerator:GSM:FREQuency 947067710"
scpi ":RFGenerator:GSM:LEVel -80"

case "Ch 700 -55dBm"
scpi ":RFGenerator:GSM:STATe ON"
scpi ":RFG:GSM:MODulation:TSEQuence:STATe OFF"
scpi ":RFGenerator:GSM:FREQuency 1842867710"
scpi ":RFGenerator:GSM:LEVel -55"

case "Ch 700 TX-IQ"
scpi ":MEASure:BLOCkdata:MSPectrum"

end select

close #1
close #2

end
```

In the above example, the scroll box allows for three different test steps. In the first two, the generator is set up. In the third step, modulation spectrum measurements are performed and displayed.

Parameter menus

The Parameter menus allow you to access the parameters relevant to asynchronous mode at a more detailed level, which need not be changed in everyday use. To access the Parameter menus, press the **Param...** softkey on the Generator menu and then press the appropriate marker tab.

The following Parameter menus are available:

- Basic parameters
- Power/time templates (PTT) (see page 81)
- Measurement limits (see page 85)
- Coupling loss (see page 90)
- Corner points (see page 94)

Basic parameters

The figure below shows the Basic Parameter menu.
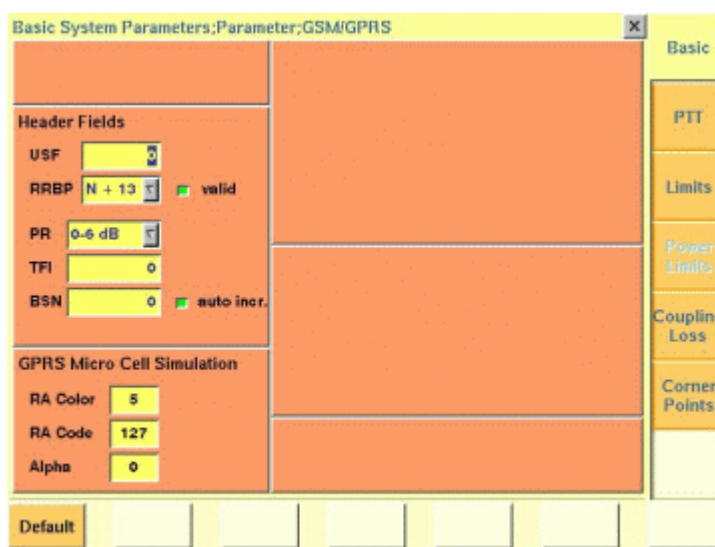
The Basic parameter menu is divided into the following sections:

- The GSM Micro Cell Simulation section provides access to the GSM system parameters broadcast in the System Information messages. The parameter fields are identical to those in the GSM environment (see "Basic system parameters" on page 76).
- The GPRS Micro Cell Simulation section provides access to GPRS system parameters broadcast in the System Information messages, especially SI13.
- The Header Fields section defines MAC or RLC header parameter values.

Input parameters The input parameters to be entered in the GPRS Micro Cell Simulation and Header Fields sections are listed below. These apply when the RF generator is set to transmit a typical GPRS signal with BCCH and packet data channel.

GPRS Micro Cell Simulation section

RA Color — Sent in the SI3 and SI4 Rest Octets. The normal use is that different different RA Color codes in adjacent cells indicate to the mobile that the cells belong to different Routing Areas. The valid range is 1...7.

RA Code — The Routing Area Code as broadcast in the SI13 Rest Octets. The valid range is 0...255.

Alpha — One of the GPRS Power Control Parameters (SI13 Rest Octets) sent also in the IA Rest Octets and RR Packet Uplink Assignment. Alpha is the coded parameter Alpha involved in the equation below. Valid entries 0...10 correspond to Alpha = 0.0...1.0, default is 10. When the mobile station transmits on a PDCH, it uses the parameter Alpha to determine its output power level Pn of the n-th burst within TDMA frame as follows:

$$Pn = \min[Gamma_0 - Gamma_{CH} - Alpha(C + 48), P_{max}]$$

where $Gamma_0$ = 39 dBm for GSM 900 and $Gamma_0$ = 36 dBm for GSM 1800/1900, $Gamma_{CH}$ is defined for each PDCH individually (e.g. during channel assignment), Alpha is defined above, C is the power level of the basis station received by the mobile, and P_{max} is the maxim output power of the mobile according to its power class.

Header Fields section — The entries in some of the GPRS header fields are user-selectable; for each TDMA frame, one PDTCH time slot is coded and duplicated into all time slots containing the PDTCH.

USF — Defines the Uplink State Flag, which can take on one of eight values either to assign a block to a particular mobile (out of seven possible), or to mark a block as unreserved. It is part of the MAC header on the PDTCH. All time slots use the same USF. The valid range is 0...7.

RRPB — The Relative Reserved Block Period field is part of the MAC header on the PDTCH, taking on one of the following values: N+13, N+17, N+21, or N+26. It marks the block to which the USF field applies relative to the currently used block.

RRBP Valid — This check box enables you to select or deselect the RRBP Valid flag in the MAC header. The flag indicate whether the above RRBP value shall or shall not be used by the mobile station. Default is "off".

PR — The Power Reduction field indicates the RF power in the downlink PDCH relative to the BCCH. It is used in downlink RLC Data Blocks. Valid entries are 0-6 dB, 8-14 dB, 16-22 dB, 24-30 dB, default is 0-6 dB.

TFI — The Temporary Flow Identifier field identifies the Temporary Block Flow TBF to which the RLC data block belongs. This parameter is used in the downlink RLC Data Block. Valid entries are in the range 0...31, default is 15.

BSN — The Block Sequence Number field is the Absolute Block Sequence Number modulo 128 of each RLC data block within the TBF. It is used to enumerate the RLC data blocks, starting with 1. Valid entries are in the range 0...127, default is 1.
This parameter can either be set to a fixed value or be increased by the 4463 GSM/GPRS System Option and 4468 EDGE System Option using the auto incr. check box.

auto incr. — If selected, the BSN is automatically increased with each new RLC data block, starting with a user-defined BSN value. Default is "on".

Tools

6

This chapter provides task-based instructions for additional tools within the 4463 GSM/GPRS System Option and 4468 EDGE System Option. Topics discussed in this chapter are as follows:

Overview

This chapter shows you the functionality in the Tools section of the user inter-face. The Tools menus can be found with a push on the Tools   function key.

The Tools menus can be divided into the following sections:

– Section "Configuration" on page 196 lets you view and change the instru-ment configuration. This includes software updates and upgrades and inter-face settings.

– Section "Utilities" on page 209 includes general functions such as a config-urable power supply and current measurements.

– "Audio measurements" on page 231 shows you how to operate the audio signal generator and analyzer.

– The section about the "HSCSD-multislot option" on page 250 details the setup and measurements for HSCSD phones.

– Section "Spectrum measurements" on page 215 gives a short overview on the spectrum meausurements available. For a detailed description refer to "RF Analyzer for spectrum" on page 47.

Configuration

With the help of these menus, you may configure the interfaces of your 4400, check whether a specific option has been installed or install options and new software releases.

This section consists of the following parts:

– "Setup" – In this menu, you may select the language of the external keyboard as well as some other settings.

– "I/O configuration" – This menu allows to set the GPIB address and the external printer connected.

– "Options" – This menu provides an overview of the options installed on this 4400 and also allows to install further options.

– "Service and software update" – In this menu, you will find all details about the software revisions installed in the various subsystems of your test set. These details are very useful when discussing problems with the Willtek product specialists. Furthermore, this menu allows to install new software releases or to return to software releases installed previously.

Access to the configuration menus

Push the **TOOLS** function key. The tool bar provides access to the configuration menus with a push on the [Config] marker tab.

Setup Push the [Setup] marker tab to gain access to this menu.
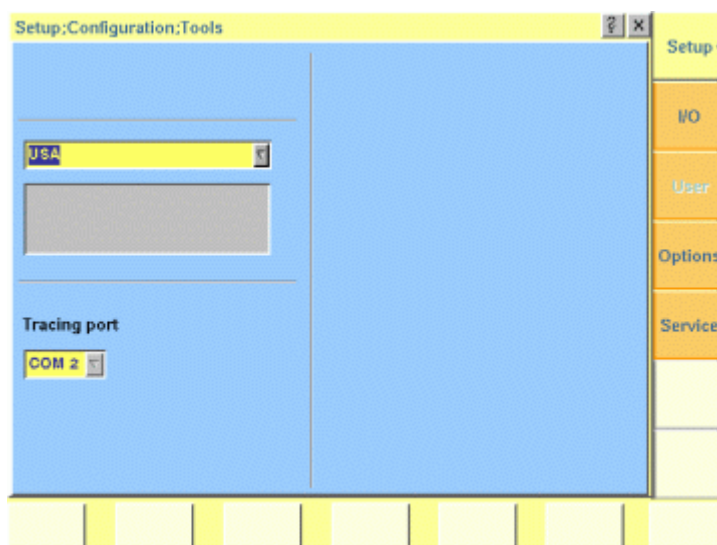
Settings of the Setup menu **External Keyboard —** This selection field allows to set the language of the external keyboard connected.
The settings possible are:
– `USA` (default)
– `BelgiumFrench`
– `BelgiumDutch`
– `CanadaFrench`
– `CanadaEnglish`
– `Denmark`
– `France`
– `Germany`
– `Italy`
– `Japan`
– `LatinAmericaSpanish`
– `LatinAmericaPortugal`
– `Netherlands`
– `Norway`
– `Portugal`
– `Spain`
– `Sweden`
– `SwissFrench`
– `SwissGerman`
– `UnitedKingdom`

> **NOTE**
>
> The language set for the external keyboard affects the mapping of special keys and characters. This may in some cases affect the "Keyboard mapping table".

> **NOTE**
>
> Any change on this selection field will require a reboot of the test set. You will be reminded by the `Please reboot` message appearing on the display field just below this selection field.

I/O configuration Push the [I/O] marker tab to gain access to this menu.
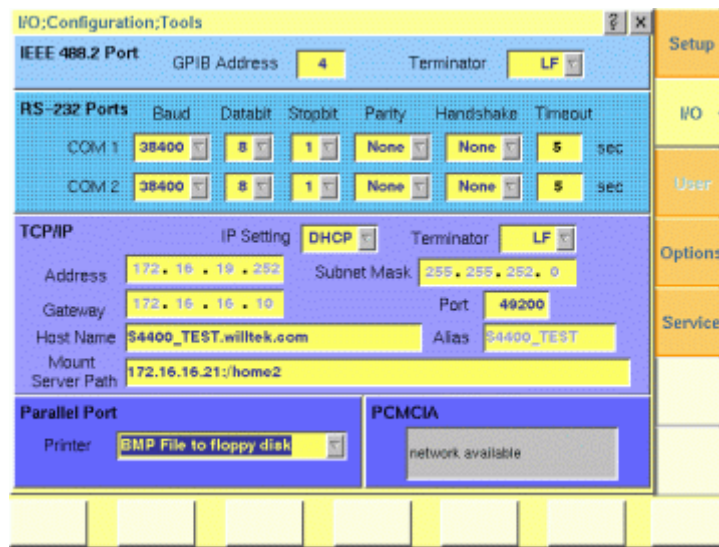
This menu consists of five areas:

- **IEEE 488.2 Port** is the area where you set the GPIB address and terminator to be used by the 4400.

- **RS-232 Ports**. These entry fields do not have a function in this software release.

- **TCP/IP**. In this area, you can configure the TCP/IP port of the 4400 and define an address on the network that you want to access from the 4400.

- **Parallel Port** provides a selection of printer drivers (or alternatively a file format) to print out the screens of the 4400.

- **PCMCIA** informs you about the PCMCIA connections currently available.

IEEE 488.2 port settings This area of the menu gives access to the following parameters:

GPIB Address — Entry field to set the GPIB address of the 4400 in case the test set is controlled externally (and not used as the GPIB controller).
Entry range: 0 to 31. Default is 4.

Terminator — Selection field to set the terminator for GPIB commands.
Settings possible are: LF (default), CR and CR/LF.

TCP/IP port settings   The 4478 TCP/IP Option provides a range of additional applications to work with the 4400 over a local area network (LAN):

– Software updates (see section "Service and software update" on page 206)

– Remote control of the 4400 using "SCPI" commands

– Access to files on the 4400 using an NFS connection

– Access to files on the network from the 4400 using an NFS connection (see sections "File menu" and "Commands for input/output handling" in the RAPID! chapter)

The following conditions must be met in order to use these features and to set up the TCP/IP parameters of the 4400:

– The 4478 TCP/IP Option must be installed. This option is available through Willtek and can be installed in the field.

– The Ethernet PC Card must be installed. This is a PCMCIA card available through Willtek. The card must be inserted upside down into the left PCMCIA card slot. Please refer to the manual of the Ethernet PC Card for more information.

If both are installed, the 4400 runs TCP/IP software and also an NFS (network file server) client/server application to allow access both from and to the 4400 over a local area network (LAN).

The following TCP/IP and NFS parameters can be configured in this area if the TCP/IP option is available and a network card has been installed:

TCP/IP		
	IP Setting DHCP ▽	Terminator LF ▽
Address 172 . 16 . 19 . 252	Subnet Mask 255 . 255 . 252 . 0	
Gateway 172 . 16 . 16 . 10		Port 49200
Host Name $4400_TEST.willtek.com		Alias $4400_TEST
Mount Server Path 172.16.16.21:/home2		

NOTE

The LAN interface card (Willtek's Ethernet PC Card) must be plugged into the PCMCIA slot BEFORE turning on the 4400 because the driver software is loaded only during the boot-up phase.

NOTE

If you want to access directories and files that reside on a computer on the network, that computer must run an NFS server software. Computers running a UNIX operating system often have NFS server software already installed. NFS server software for Windows- and for MacOS-based computers is available as a third-party product.

> **NOTE**
>
> If you want to access directories and files on the 4400 from a remote com-
> puter, that computer must run an NFS client software. Most computers run-
> ning a UNIX operating system have NFS client software already installed. NFS
> client software for Windows- and for MacOS-based computers is available as
> a third-party product.

Terminator — This selection field is used to set the terminator for SCPI
commands when controlling the 4400 via LAN. The allowable settings are: **LF**
(default), **CR** and **CR/LF**.

Address — If IP addressing is set to fixed, enter a valid IP address for the 4400
in this field, in the format **a.b.c.d**; **a** through **d** are numbers in the range 0
to 255. The address must not already be in use by another device. Refer to the
administrator of your local area network for the assignment of a valid and acces-
sible IP address within your local area network.
If IP addressing is set to DHCP, the IP address of the 4400 is automatically derived
from a DHCP server.

> **NOTE**
>
> If you enter a new address you need to reboot the 4400 for the changes to
> take effect.

Port — Port number in the 4400 for access to the 4400 when controlling the
instrument from a remote computer via LAN using the SCPI command set. The
default is 49200. For successful remote control via LAN, you will have to use the
same port number in the application controlling the 4400 as defined in this input
field.

Mount Server Path — You can access a server disk on the network from certain
applications on the 4400 if you enter a valid network address for that disk here.
A valid address consists of a server IP address followed by a colon and a path
(directory or folder) that is accessible from the 4400 over the network. A typical
example for a server path is **172.16.16.21:/disk3/4400_sw**.
Typical applications include software updates and loading files from and to the
network using the RAPID! environment.
In the File manager, the mounted disk will appear as **/io/server**.

> **NOTE**
>
> The 4400 is not aware of the address for your local domain name server,
> therefore you have to provide the IP address (in the a.b.c.d format) rather than
> a symbolic computer name.

> **NOTE**
>
> To access a remote computer, that computer must grant access rights for the
> appropriate service, e.g. if you want to write data to the disk on the network
> the operating system on that computer must be set up to grant write access
> to all users for the respective directory on that disk.

> **NOTE**
>
> This field allows you to mount only one disk. If you need to access server disks from a RAPID! program or from a remote control application (via GPIB or TCP/IP), you can mount more than one disk from your application program.

TCP/IP Troubleshooting

1 The network connection does not appear to work at all.

– Ensure that the network card is installed when you switch on the 4400. You can remove and reenter the same PCMCIA card once the 4400 has booted.

– Check if the 4400 sounds a double-beep while booting. If it does, it found the network card; if not, the PCMCIA card may be defective.

– Check if the cables are tightly connected to the PCMCIA card and to the Ethernet wall outlet.

– Ensure that the PC supports the TCP/IP protocol stack. When a network card is installed, some versions of the Microsoft Windows operating system default to another protocol such as NETBEUI. Windows can run several protocol stacks simultaneously, so installing the TCP/IP stack usually does not affect other network connections and applications.

– Check if the IP address is valid and not already in use by another device on the network. You can check this with your computer if it is on the same network as the 4400: On a PC running the Microsoft Windows operating system, open a command shell box (usually under Start-Programs-Accessories) and enter the following command line: `ping <address>` where <address> is the IP address of your 4400.
The ping should time out if the 4400 is switched off, but there should be responses from your 4400 if it is up and running. If there are pings while the 4400 is switched off, there is another device (such as a PC or another 4400) on the network; there must only be one device per address on the network.
If there is no ping even when the 4400 is switched on, the address is not a valid IP address in this network or network segment, or the 4400 is not properly connected to the network (check cabling).

2 I can enter a new IP address, but the 4400 does not respond to messages over the Ethernet connection.
After making changes to the IP parameters, they are only used after rebooting the 4400.

3 Although I have entered an IP address for the 4400 and a server path, I do not see the mounted device in the list of directories (e.g. when installing a new software version).

– Check if the IP address is valid and not already in use by another device on the network. You can check this with your computer if it is on the same network as the 4400: On a PC running the Microsoft Windows operating system, open a command shell box (usually under Start-Programs-Accessories) and enter the following command line: `ping <address>` where <address> is the IP address of your 4400.
The ping should time out if the 4400 is switched off, but there should be responses from your 4400 if it is up and running. If there are pings while the 4400 is switched off, there is another device (such as a PC or another 4400) on the network. There must only be one device per address on the

network.
If there is no ping even when the 4400 is switched on, the address is not a valid IP address in this network or network segment, or the 4400 is not properly connected to the network (check cabling).

– Check if the remote PC runs an NFS server software. UNIX-like operating systems typically include an NFS server; applications for Windows are commercially available.

Parallel port settings
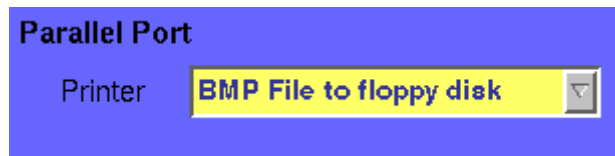
Printer — Selection field. Selects the type of the external printer, connected to the 4400's parallel port ("CENTRONICS").
The possible settings are:
- `HP DeskJet 400 (default)`
- `HP DeskJet 680`
- `HP Laserjet`
- `Canon`
- `Epson Stylus Color`
- `Epson Stylus Color II`
- `Epson Stylus Color IIs`
- `Epson Stylus Color 200`
- `Epson Stylus Color 400`
- `Epson Stylus Color 500`
- `Epson Stylus Color 600`
- `Epson Stylus Color 800`
- `Epson Stylus Color 1500`
- `Epson Stylus Color 1520`
- `Epson Stylus Color 3000`
- `Epson Stylus Pro`
- `Epson Stylus Pro XL`
- `Epson Stylus Pro XL+`
- `BMP File to floppy disk`

Note: Setting `BMP File to floppy disk` will output the current screen as a bit-map file (*.BMP) on a disk inserted into the 4400's disk drive. A prompt will appear on-screen when the 'printing to disk' process starts. The printing can be aborted by pushing the **ESCAPE** function key or the corresponding key of the external keyboard as long as the prompt is visible.
The bit-map file may then be used with standard word processing or DTP software.
The size of a single bit-map file is approx. 305 kB; therefore four files will fit on a disk.

Options   Push the [Options] marker tab to gain access to this menu.
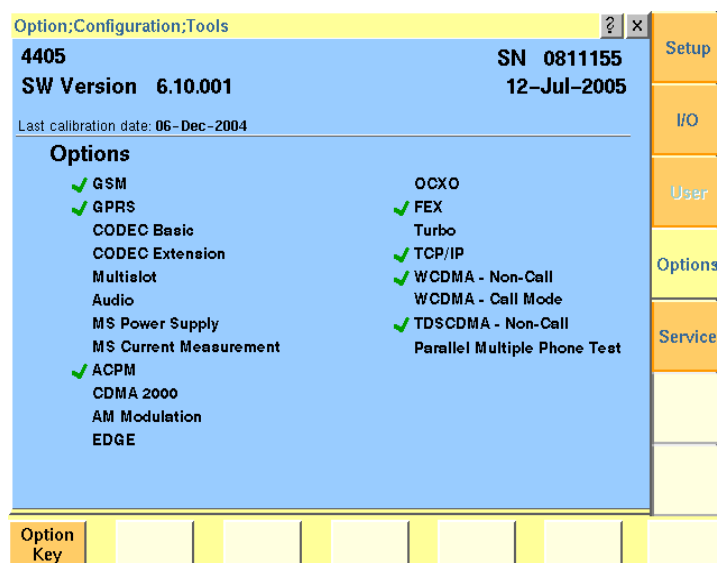
This menu consists of two areas:

– The upper area of the menu is called the basic info area. Here you will find the basic information about your test set, like the software release currently installed.

– `Options` is the larger area of the menu that informs you about all options available and the ones installed. See "Accessories and options" on page 29 for more details on available options.

The **Option Key** softkey is used to install additional options (see below).

Basic info area   This area of the menu gives the following basic information:
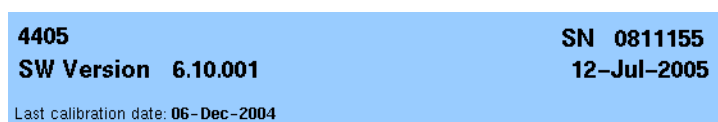
4400 — In the upper left corner of this area, the 4400 model is indicated (4400, 4403, 4405 or 4407).

SN 0811155 — In the upper right corner of this area, the serial number of this test set is displayed.

> **NOTE**
>
> The availability of an option may be dependent on the serial number.

SW Version 6.10.001 — Here, the software version currently in use on this test set is displayed.

12–Jul–2005 — This is the release date of the installed software version.

Last calibration date: 06–Dec–2004 — Here, the date of the last calibration of the test set will be displayed.
If a calibration date of the whole unit is not found, the date of the last calibration of the RF section is indicated.

Options The options area provides a quick overview of all options installed.

Options

✓ GSM	OCXO
✓ GPRS	✓ FEX
CODEC Basic	Turbo
CODEC Extension	✓ TCP/IP
Multislot	✓ WCDMA - Non-Call
Audio	WCDMA - Call Mode
MS Power Supply	✓ TDSCDMA - Non-Call
MS Current Measurement	Parallel Multiple Phone Test
✓ ACPM	
CDMA 2000	
AM Modulation	
EDGE	
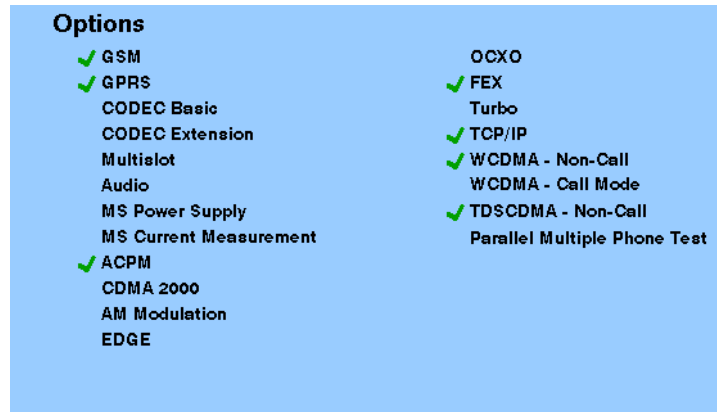
GSM — When a tick is shown in front of this list entry, the GSM System Option software is installed.

GPRS — When a tick is shown in front of this list entry, the GPRS System Option software is installed.

CODEC Basic — When a tick is shown in front of this list entry, the Basic CODEC Option is installed.

CODEC Extension — When a tick is shown in front of this list entry, the CODEC Extension Option is installed.

Multislot — When a tick is shown in front of this list entry, the HSCSD-Multislot Option is installed.

Audio — When a tick is shown in front of this list entry, the Audio Option is installed.

MS Power Supply — When a tick is shown in front of this list entry, the MS Power Supply Option is installed.

MS Current Measurement — When a tick is shown in front of this list entry, the MS Current Measurement Option is installed.

ACPM — When a tick is shown in front of this list entry, the ACPM Option for special spectrum measurements is installed. This is always the case with the 4400M where ACPM is a standard feature. Note that ACPM is not an option in the CDMA2000 1xRTT Option.

CDMA 2000 — When a tick is shown in front of this list entry, the CDMA2000 1xRTT Option is installed.

AM Modulation — When a tick is shown in front of this list entry, the AM Signal Generator Option is installed.

EDGE — When a tick is shown in front of this list entry, the EDGE System Option software is installed.

OCXO — When a tick is shown in front of this list entry, the OCXO Option is installed to provide a high-precision time base.

FEX — When a tick is shown in front of this list entry, the Frequency range Extension Option is installed.

Turbo — When a tick is shown in front of this list entry, the Turbo Option is installed.

TCP/IP — When a tick is shown in front of this list entry, the TCP/IP Option is installed.

WCDMA Non-Call — When a tick is shown in front of this list entry, the WCDMA Non-Call Mode Option is installed.

WCDMA Call Mode — When a tick is shown in front of this list entry, the WCDMA Call Mode Option is installed.

TD-SCDMA Non-Call — When a tick is shown in front of this list entry, the TD-SCDMA Non-Call Mode Option is installed.

Parallell Multiphone Test — When a tick is shown in front of this list entry, the Parallel Multiphone Test Option is installed.

Installing additional options

Some of the options of the 4400 require both software and hardware (like e.g. the high-precision time base option or OCXO). These options can only be installed by the Willtek product specialists.
Options that are software-based only (like ACPM for instance) may be installed on-site. The only thing required then is a special code available from Willtek. When you have the code at hands, follow the instructions below:

1   Push this softkey. **Option Key**

2   The `Input Option Key` prompt will appear on the screen.

3   Enter the option key received from Willtek.

4   Confirm the option key entered by pushing the **ENTER** function key or the corresponding key on the external keyboard.

5   If the code is recognized by the 4400, it will display an ACCEPT message. After leaving the entry box with the ESCape key, the tick in front of the corresponding option will appear and the full functionality of the option will be at your disposal.

6   In case the option key entered was not recognized by the 4400, you will see the `NO VALID KEY!` message. Try to enter the correct option key again.

Service and software update

Push the [Service] marker tab to gain access to this menu.
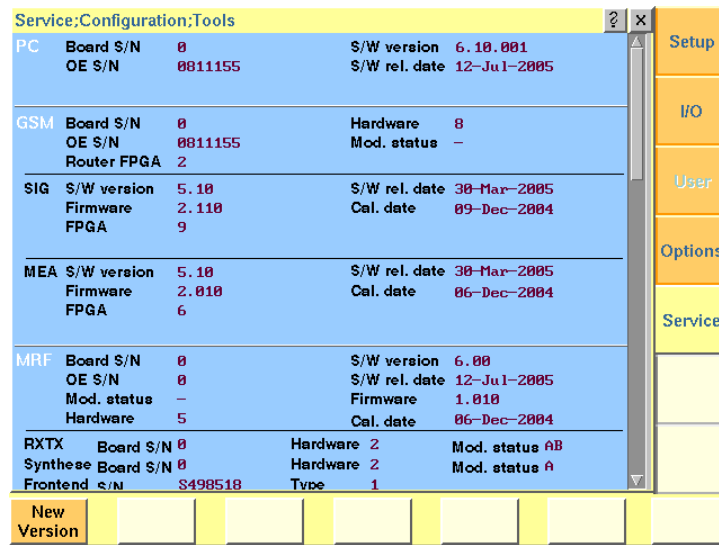
This menu provides all relevant information on the hardware and software revisions of the various subsystems of your 4400.

The information on display here is very important when tracing problems or discussing upgrade possibilities with the Willtek product specialists.

Changing the version of the system software

Willtek seeks to permanently improve its products and makes new software versions available on the Internet at www.willtek.com. Several different methods exist to upgrade the software of your 4400: From floppy disks, from a remote PC over the LAN (TCP/IP required) and from a remote PC over GPIB.

⚠️ **HAZARD**

Be very careful using the functions provided in this menu. If used the wrong way, functions described below may leave your test set in a condition where it is not operable anymore.
Never 'play' with this menu!

A push on the **New Version** softkey provides access to the following menu:
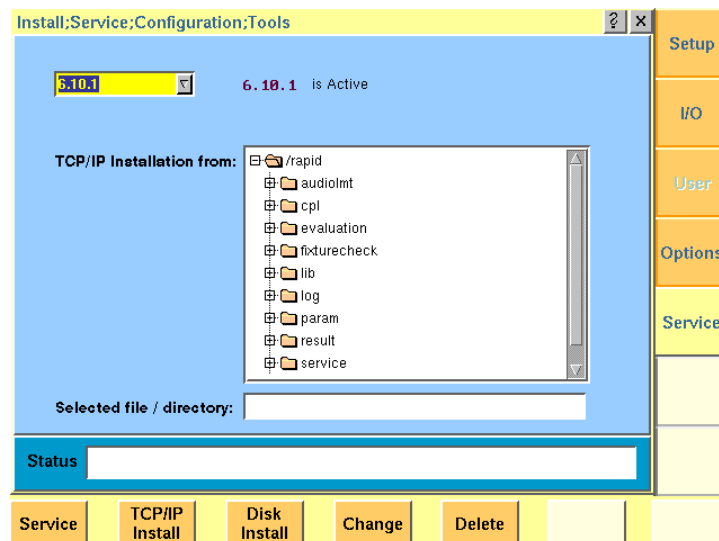
The display field in the upper middle of the menu (**6.10.1 is Active** in this example) indicates the software release that is currently in use.

The selection field to the left of the display field allows you to select a different software release that was previously installed on this test set. This will usually be an older software version.

NOTE

The 4400 allows a maximum of five software releases to be stored internally. In case you try to update a test set where five software releases have been stored already, you will be prompted to delete a software release first.

CAUTION

After installing version 5.10 version 5.00 must be deleted as switching back to the old version can in this case cause problems.

The file selection box **TCP/IP Installation from:** allows you to select a file on the network from which to install new software. The file with the software upgrade usually has the file name extension **.tar**. The box displays the directories and files available on the mounted server disk. The server disk can be mounted in the **Tools > Config > I/O** menu.

NOTE

The network computer with the server disk must be an NFS server.

The **Status** area of the menu is where the 4400 will display additional information, comments and messages.

The softkeys of this menu

Service	Closes the current menu and displays the service menu (toggle). The softkey will change to **New Version**.
TCP/IP Install	A push on this softkey starts the **update routine via LAN (TCP/IP)**. The 4400 will load and install the file selected in the file selector box. Please follow the instructions shown on the screen of the 4400. **Note:** NEVER close the window or switch off the 4400 while the 4400 is updating the system (i.e. after copying software from the network onto the internal hard disk and before the EPLD update is completed). Otherwise the instrument will not be usable anymore and needs to go back to the factory or to a Willtek service center.

Disk Install

A push on this softkey starts the **update routine from floppy disk**. Please follow the instructions shown on the screen of the 4400.

Notes:

– To return from the update routine without updating, close the window on-screen with the external mouse and confirm the prompt with **Yes**.

– NEVER close the window or switch off the 4400 while the 4400 is updating the system (i.e. after copying software from floppy disk onto the internal hard disk and before the EPLD update is completed). Otherwise the instrument will not be usable anymore and needs to go back to the factory or to a Willtek service center.

Change

This softkey can be used to change from the current software version to a previously installed version.

First, select the software release to change to on the selection field of this menu.

Changing to another softkey takes less than one minute on most systems, but the software must be rebooted.

Before pushing this softkey, consider the consequences: Software releases lower than 2.86, for instance, will allow no return to the current release state.

Note: After installing version 5.10 version 5.00 must be deleted as switching back to the old version can in this case cause problems.

Delete

If five software versions are already installed on the 4400's hard disk, it will prompt you to delete an older version first when you try to install yet another version.

First, select the software release to be deleted on the selection field of this menu. Before pushing this softkey, consider the consequences.

After a software release has been deleted, it must be installed anew if you want to work with it again.

A currently used software version cannot be deleted. If you want to delete the current version, change to another version first.

Initiating a software update from a remote PC

Willtek also offers a tool to update the 4400 software from a PC over the GPIB. This requires a GPIB interface in the PC. The file to download and a Windows-based program to perform the software download are available in the download section on www.willtek.com.

NOTE

The download program requires a National Instruments GPIB card.

Utilities

The menus of this section provide useful tools for debugging when the 4400 is used in a system environment.

This section consists of the following parts:

– "I/O trace for GPIB communications" – allows to monitor communication on the GPIB in great detail.

– "Info trace" – stores internal messages of the system and is helpful when discussing problems with the Willtek product specialists.

– "MS power supply" – the MS Power Supply option provides the mobile phone with supply voltage; the Current Measurement option helps identify problems that are caused by shortcuts on the printed circuit board of the phone.

Access to the Utilities menus

Push the **TOOLS** function key from the Welcome menu. The tool bar provides access to the utilities menus with a push on the [Utilities] marker tab.

I/O trace for GPIB communications

This menu allows for monitoring the communication on the GPIB, e.g. to debug a new PC program for remote control of the 4400.
Push the [I/O Trace] marker tab to gain access to this menu.
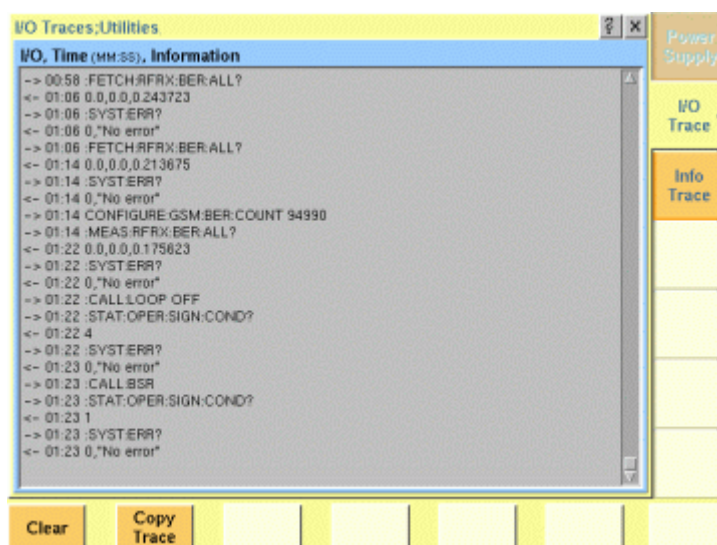
This menu consists of an area to show all the commands and responses going back and forth between a controlling device (such as a PC) and the 4400. This communications display is called "trace" and is written to both the screen and a trace file.

NOTE

The trace is generated only as long as the screen is active, otherwise no trace is generated (or the existing trace will not be updated with new information).

NOTE

Generating the trace consumes more processing power from the 4400 i.e. the 4400 may work slower than without tracing.

NOTE

If you don't need the trace for debugging purposes, it is strongly recommended to go back to the Welcome menu before starting remote control operation.

The softkeys of this menu

Clear	Resets the entire I/O Trace menu (all entries in all areas of the menu will be cleared).
Copy Trace	Before pushing this softkey, insert a disk in the 4400's disk drive as a push on this softkey copies the entire I/O Trace to the disk. A text file named GPIBPROT.TXT will be generated, containing the I/O Trace in ASCII format. If the I/O Trace is empty, the file will be empty as well.

Info trace   Push the [Info Trace] marker tab to gain access to this menu.
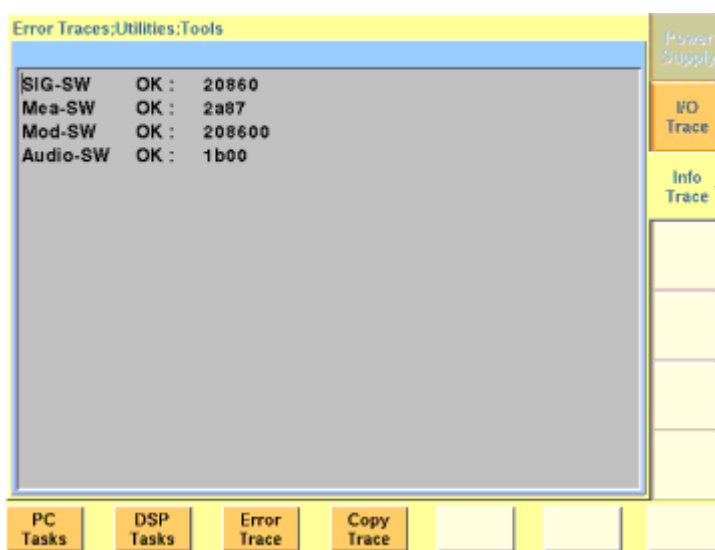
This menu provides detailed information about internal procedures and messages of the 4400.
It does not contain any user-serviceable settings or entry fields.
Its main application is internal debugging at Willtek. It may, however, also provide useful information for the Willtek product specialists when tracing a problem.

The softkeys of this menu

PC Tasks — Shows the trace of the procedures, running on the internal PC.

DSP Tasks — Shows the trace of the procedures, running on the internal DSP.

Error Trace — Shows internal messages and error trace.

Copy Trace — This softkey can be used to copy the Error Trace onto a floppy disk.
Before pushing this softkey, insert a disk in the 4400's disk drive. A push on this softkey copies the internal message and error trace onto the floppy disk. A text file named INFO-DAT.TXT will be created, containing the internal message and error trace in ASCII format. If the trace is empty, the file will be empty as well.
The Error Trace file may be requested by Willtek Product Specialists when reporting a problem.

MS power supply

Push the [Power Supply] marker tab of the Utilities section to gain access to this menu. It offers you the controls and measurements for two options: the MS Power Supply Option and the Current Measurement Option.

The MS Power Supply is an option to the 4400, therefore this screen is accessible only if the MS Power Supply Option is installed.

The MS Power Supply Option is a prerequisite for the Current Measurement Option. The Current Measurement Option shows you the current that the mobile phone draws from the battery simulated by the 4400.

The measurement results from the Current Measurement Option are visible only if this option is installed.
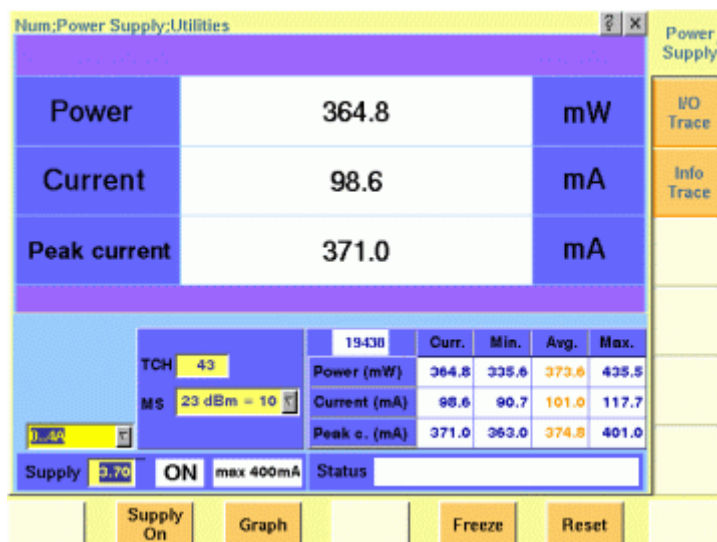
MS Power Supply Option This menu allows you to select and switch on/off the required supply voltage for the mobile phone, for use with the MS Power Supply option of the 4400. The MS Power Supply options is meant to replace a battery e.g. in environments where the original battery is not yet connected to the mobile (such as in manufacturing lines) or where faults must be traced back to either the phone or the battery (such as in service). Moreover, the MS Power Supply can be used to test the effect of over- or under-voltage on RF transmitter, RF receiver and audio parts.

Connection: The MS Power Supply connector can be found at the left-hand side of the 4400 front panel, below the floppy disk drive. The MS Power Suppy option comes with a cable of one meter length. One end of the cable can be plugged into the 4400, while the other end is open-ended. The open ends can be used to connect the power supply to a fixture holding the mobile phone. Willtek delivers the cable with open ends because these fixtures are manufacturer-specific and hence open ends are most universal.

The Supply entry field can be used to select a supply voltage in the range 0 to approximately 11 V DC, in steps of 0.05 V (i.e. 50 mV). The option is specified from 0 to 10 V; higher voltages may or may not be achievable.
A status field next to the supply voltage entry field indicates the current status of the power supply, i.e. either ON or OFF. A softkey is available to turn the 'supply on' or 'supply off', respectively.

> **NOTE**
>
> For safety reasons, the supply voltage is always off at power-on. So the power supply has to be switched on when required.

> **NOTE**
>
> If the power supply is short-circuited, the supply voltage and current goes down. When the short circuit is removed again, the voltage will come back after a few seconds (PTC resistor).

Current Measurement Option If installed, this option displays the current drawn from the simulated battery.

The menu can show the measurement results in two different modes: a numerical-only display and a combined graphical and numerical display. In both cases, the lower results section is the same, indicating a statistical representation of the results for measured power, average and peak current.

The upper results section varies depending on the selected display mode; the mode can be selected with a softkey.

– In numerical-only mode, the upper results section shows the numerical results for power and current from the latest measurement.

– In graphical mode, the upper results section displays the shape of the current over time for one TDMA frame.

Selecting the display mode and display properties

The numerical display is the default mode. To switch to the combined graphical/numerical mode, push the `Graph` softkey. As a result, the current versus time display appears and the softkey description changes to "Numeric".
Push this softkey again to return to numerical-only value.

In combined graphical/numerical mode, the menu shows a graph with the current versus time. The scale of the vertical axis is mA and the horizontal axis scaling is in ms (milliseconds).
Two cursors and a cursor readout area allow you to read the measurement values.

The graphical display resolution can be changed as well: Use the scroll field in the numerical results area to change the vertical resolution.

> **NOTE**
>
> The Duration field is not available in this software version.
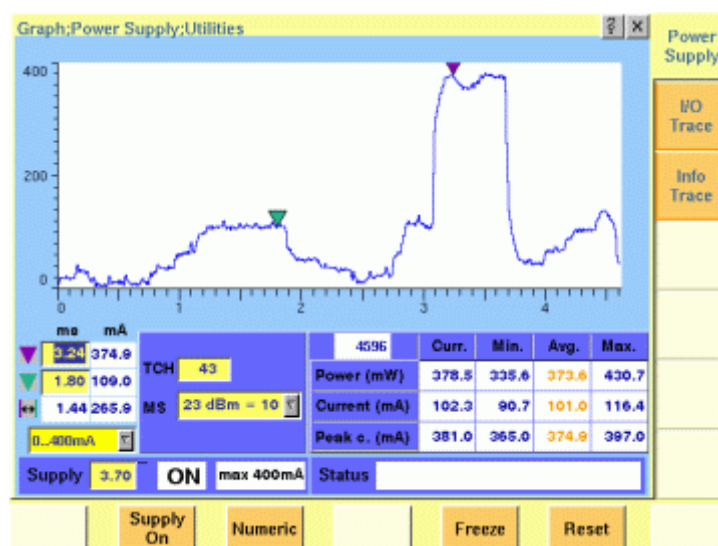
The measurement and display range covers a complete TDMA frame from the start of a downlink frame. With a traffic channel active on time slot 3 and an offset of three time slots between downlink and uplink, the highest current level can be expected at $(3+3) * 0.54738$ ms $= 3.28$ ms.
Other current peaks may result from an active receive/demodulation process.

In the measurement example above, a period of low current can be identified in the beginning. After 1 ms, the mobile phone sets up its receiver for reception of the downlink slot. From 1.8 ms to 3 ms, the demodulated data are decoded, and the transmitter prepares for the uplink slot. Transmission of the uplink slot follows, requiring a lot of current from the mobile phone; the absolute value depends on the selected uplink power level. The final phase of the frame is characterized by preparations for the next uplink slot transmission, for example interleaving in the channel coder.

Understanding result fields

The difference between current, minimum, maximum and average values is explained in section "Statistical data evaluation" on page 244.

Power — This field shows the average power drawn from the simulated battery, measured in mW. It is calculated from the selected supply voltage and the measured current.

Current — The Current Measurement Option measures the current that the mobile draws from the MS Power Supply outlet. The result is averaged over one TDMA frame and displayed in mA.

Peak c. — This field indicates the maximum current that has been detected during the last TDMA frame. The peak current result is shown in mA.

Changing the RF parameters

While measuring the current consumption, you can also switch the RF channel or the transmitted RF power to evaluate the current consumption under different parameter settings.
Use the input fields TCH to change to a different channel.
Use the MS scroll field to change the transmit power level.

Spectrum measurements

The group of spectrum measurements available in the Spectrum menu offers functions for measuring the RF output spectrum emission. These functions are used to measure the parameters for assessing the quantity of power that leaks outside the assigned radio channel, i.e. the off-carrier power. If this power value exceeds certain limits, interference with neighbouring channels may increase. Furthermore, system capacity may be affected. Spectrum measurements for TD-SCDMA include Occupied Bandwidth (OBW), Adjacent Channel Leakage Power Ratio (ACLR) and Spectrum Emission Mask (SEM).

In order to get access to the Spectrum menu push the **TOOLS** function key. You can also select the [Spectrum ...] marker tab while working in Generator/ Analyzer mode. For a detailed description of the spectrum measurements available refer to "RF Analyzer for spectrum" on page 47

Spectrum measurements

The spectrum analyzer menus provide test capabilities to evaluate the RF signals received by the 4400 in the frequency domain.

The measurement of the modulation spectrum is a standard on all the 4400s. The **A**djacent **C**hannel **P**ower **M**eter spectrum measurements on both the spectrum due to modulation and due to switching transients are standard features in the 4400M and optional to all other instruments of the 4400 series.

This section consists of the following parts:

– "Modulation spectrum measurements" – This section gives an overview of the measurements of the spectrum due to modulation. This measurement capability is a standard on all the 4400.

– "ACPM modulation spectrum measurements" – Here you will find all information regarding ACPM measurements of the spectrum due to **modulation** and wideband noise. This measurement capability is an option except for the 4400M where it is a standard feature.

– "ACPM transient spectrum measurements" – Here you will find all information regarding ACPM measurements of the spectrum due to **switching transients**. This measurement capability is an option except for the 4400M where it is a standard feature.

– "Softkeys" – This section provides an overview of the softkeys used in the spectrum measurement menus.

– "Display markers" – are used to take measurements in the graphic displays.

– "Parameters and status" – This section provides an overview of the parameter and status display fields.

Access to the spectrum measurement menus

– Spectrum measurements are possible in all relevant operating states of the 4400.

– For tests in asynchronous mode, push the **TOOL** function key from the generator and analyzer menus. The tool bar will provide access to the spectrum measurement menus with a push on the [Spectrum] marker tab.

– For spectrum measurements on a **GSM traffic channel**, select marker tab [GSM] and then push the **Calls & Meas.** softkey from the Welcome menu. Now, originate a call (either from the mobile under test or from the base station). When the call is active, push the **TOOLS** function key. The tool bar will provide access to the spectrum measurement menus with a push on the [Spectrum] marker tab.

– During multislot tests, access to the spectrum measurement menus is provided by the **Spectrum** softkey.

Modulation spectrum measurements

The measurement of the spectrum due to modulation and wideband noise is a useful tool for

– adjusting the mobile's I/Q modulator, and

– detecting side waves of the carrier or spurious emissions.

– The menu of the modulation spectrum analyzer provides a graphical display of the RF input signal's spectrum. To call the menu, simply push the [Mod. spectr.] marker tab.
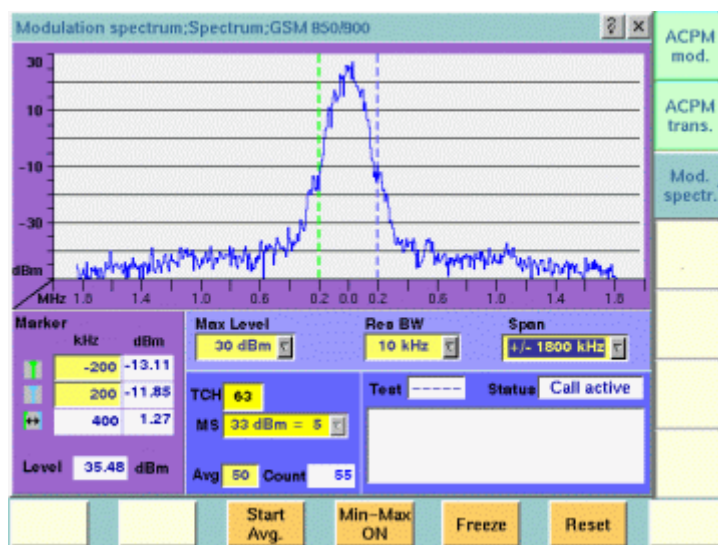
On first sight The mobile's modulation spectrum is displayed graphically in the menu's live display. The live display is updated continuously. To read the exact measurement at any position, simply use the display markers.
The current main "Parameters and status" are displayed in the lower right corner of the menu.
The softkeys provide access to important tools of the spectrum analyzer.

The Spectrum Live Display The live display is updated continuously (as long as it has not been frozen).

The vertical axis of the spectrum live display is in dBm; the horizontal axis in MHz.
The selection field `Max Level` allows to set the maximum power level to optimize the vertical resolution.
Display field `dBm` shows the maximum RF power level measured on the center frequency.
Selection fields `Res BW` and `Span` allow to set the resolution and frequency span for the horizontal axis of the live display.

Max Level — This selection field allows to set the RF power level that is represented by the top horizontal line of the live display.
Selections possible are: from +40 dBm to –40 dBm in steps of 10 dB.

NOTE

The maximum RF power level measured at the center frequency is displayed on the `dBm` display field just right to this entry field.

> **NOTE**
>
> The value selected here should be equal or slightly higher than the expected maximum RF power level on the measurement frequency in order to avoid overload distortion and to achieve maximum vertical resolution.

Res BW — Selection field to set the horizontal resolution.
Selections possible are: `2 kHz`, `5 kHz`, `10 kHz` and `30 kHz`.

> **NOTE**
>
> The resolution settings available depend from the span of the live display. The following combinations are available:
> – For the ±200 kHz and the ±500 kHz span, resolution bandwidths of 2, 5 and 10 kHz are available.
> – For the ±1800 kHz span, resolution bandwidths of 10 and 30 kHz are available.

> **NOTE**
>
> A high resolution (i.e. a small value selected) will provide many details while a low resolution (i.e. a large value selected) will provide a smooth curve.

Span — Selection field to set the frequency range shown in the live display.
Selections possible are: `±200 kHz`, `±500 kHz` and `±1800 kHz`.

> **NOTE**
>
> The selections possible for `Res BW` are dependent from the current setting of this selection field:
> – For the ±200 kHz and the ±500 kHz span, resolution bandwidths of 2, 5 and 10 kHz are available.
> – For the ±1800 kHz span, resolution bandwidths of 10 and 30 kHz are available.

Setting the center frequency

The center frequency of the spectrum analyzer is the frequency of the uplink carrier.
The center frequency cannot be modified in the RF spectrum analyzer menus.

– In synchronous mode (i.e. when a traffic channel has been set up), this frequency is given by the traffic channel number used and displayed on-screen on the `TCH` display field.
   Example: TCH channel number 45 equals a center frequency of 889.0 MHz.
– In asynchronous mode (i.e. when there is no signaling exchange between the mobile under test and the 4400), the center frequency is identical to the frequency set on the analyzer frequency entry field of the Analyzer menus.

Setting the channel and the power level

The channel number and the power level can be changed in call (synchronous) mode.

Use the **TCH** field to change the frequency. The 4400 will initiate an intra-cell handover.

Use the **MS** scroll field to change the power level at which the mobile phone is transmitting.

Reading the overall power level

The overall power is displayed in the lower left corner, below the marker readings.

Averaging the Modulation Spectrum

The 4400 provides a useful tool to smoothen the RF modulation spectrum curve: the averaging function.
This means that a user-definable number of the continuous measurements will be taken and the single measurement results will be averaged. The result will be a smooth curve that shows characteristics of the spectrum curve better than the single measurements.

Working with the averaging tool

– Enter the averaging tool by pushing the |Start Avg.| softkey.

– The 4400 will now start the averaging process, using the number of measurements set on the Avg entry field.

– As long as the number of measurements taken is less than the number entered on the Avg entry field, the number of the current measurement will be displayed in red on the Count display field.

– As soon as the number of measurements taken is equal or larger than the number of measurements specified for the averaging process, the number on the Count display field will turn blue.

Important to note

– The entry range for the Avg entry field is from 1 to 200.

– The Avg entry field is not accessible unless the averaging process has been started.

– As soon as you change the current setting on the Avg entry field, the averaging process will be reset. This means that all previous measurements will be discharged and the series of measurements will be started anew.

– A reset of the averaging process can also be achieved by pushing the |Reset| softkey.

– When the number of measurements taken (display field Count) exceeds the number of measurements specified for the averaging process (Avg entry field), always the last Avg measurements will be averaged and displayed in the live display.

– In case the live display is frozen and unfrozen (using the appropriate softkey) during an averaging process, the averaging process will not be interrupted but will continue in the background.

– Even when the live display has been frozen, a reset of the averaging is possible, pushing the |Reset| softkey.

Viewing the maximum deviation

The spectrum live display shows the current or the averaged spectrum of the signal. In some cases it can be useful to check the maximum deviation from the current or the average. Push **|Min-Max ON|** for this purpose; the 4400 then displays two additional curves in red, visualizing the minimum and the maximum for each frequency point as measured since the start of the test.

Push **|Reset|** to erase previous minima and maxima and to start the measurements all over.

Push **|Min-Max OFF|** to resume measurements without the deviation display.
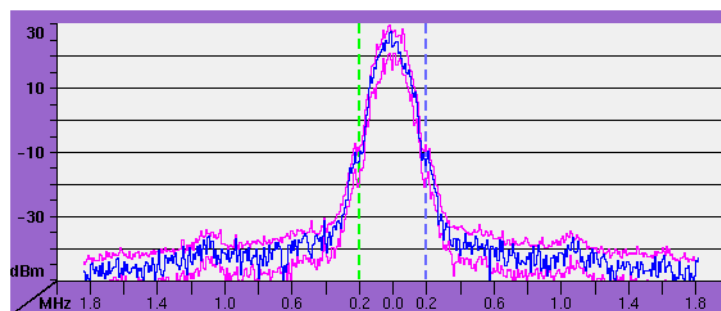
ACPM modulation spectrum measurements

The measurement of the ACPM spectrum due to modulation and wideband noise is used to check whether the spectrum of the RF signal transmitted by the mobile under test is in accordance with the appropriate GSM specifications (see below for details).
The purpose of this measurement is to verify that the mobile's modulator does not create sideband emissions that would then disturb transmission on adjacent traffic channels.
This measurement will test the following internal stages of the mobile under test:

– channel synthesizer,

– Gaussian baseband filter, and the

– I/Q modulator,

The following defects of the mobile might be detected with this measurement:

– neighbouring waves of the synthesizer and

– asymmetry of the I/Q modulator.

NOTE

Many internal stages of the mobile may also be checked using the phase and frequency error measurement.

To call the menu, simply push the [ACPM mod.] marker tab.
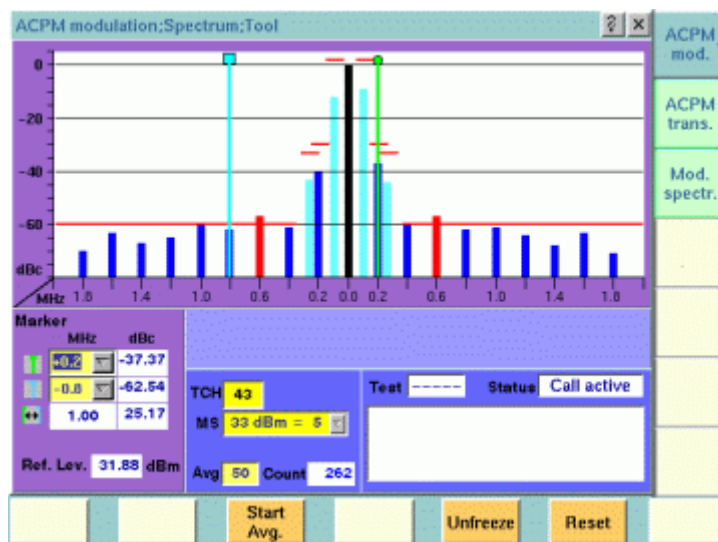
On first sight The mobile's modulation spectrum is displayed graphically for the discrete frequencies specified by the appropriate GSM specification in the menu's live display. The live display is updated continuously. To read the exact measurement at any position, simply use the display markers.

The measurement results shown are relative to the level at the center frequency. The level display at the lower left of the menu displays the absolute level at the center. This allows you to calculate the absolute values at each of the displayed frequencies.

The current main "Parameters and status" are displayed in the lower right corner of the menu.

The softkeys provide access to important tools of the spectrum analyzer.

A glance into the GSM specifications The ACPM measurements refer to section 4.2 (Output RF Spectrum) of specification GSM 11.10 (Mobile conformance specification; Radio transmission and reception).

Section 4.2.1 specifies various measurements for the 'spectrum due to modulation and wide band noise'.

The most important measurements are through a 30 kHz filter (resolution bandwidth; 100 kHz video bandwidth) at offsets of ±100 kHz, ±250 kHz and all multiples of ±200 kHz from the carrier up to (and including) ±1800 kHz.

The signal is measured over the useful part of the burst (i.e. **not** including the rising and falling edges of the burst), also excluding the midamble (i.e. the training sequence).

The GSM specification then specifies that the single measurement results shall be averaged over 200 bursts and displayed in relation to the RF power level measured (with a 30 kHz bandwidth) at the carrier frequency.

NOTE

The 'carrier frequency' is the frequency of the TCH used for this measurement.

NOTE

The upper limits for the single frequencies vary between GSM 900 and GSM 1800. The 4400 will take care of that automatically.

Understanding the measurement principle

The measurement is performed in the time domain using the zero-frequency-span of the spectrum analyzer. The 4400 is set to a resolution bandwidth of 30 kHz and the (internal) averaging function is activated. A time gate ensures that the measurement is only performed during the useful part of the burst. A reference measurement at the carrier frequency (= TCH in synchronous mode) is performed. The RF power level measured is called the **reference level** and thus equals 0 dBc. The 4400 shows the measurement result of this reference measurement in the menu.

Level measurements relative to the carrier frequency are performed at ±100 kHz, ±200 kHz, ±250 kHz, ±400 kHz, ±600 kHz, ±800 kHz, ±1000 kHz, ±1200 kHz, ±1400 kHz, ±1600 kHz and ±1800 kHz offset (by default over ten bursts each) through a 30 kHz bandwidth filter. The measurement results are then displayed as vertical bars in relation to the reference level (in dBc).

The spectrum live display

The live display is updated continuously (as long as it has not been frozen).
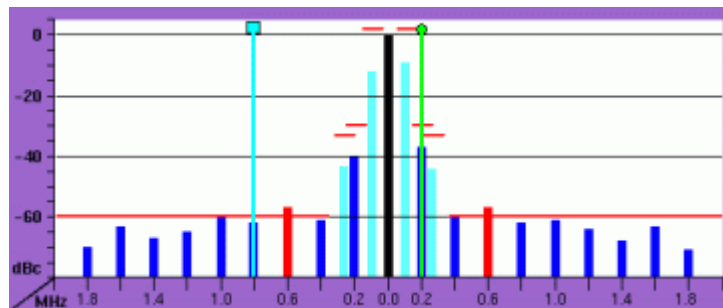
The vertical axis of the spectrum live display is in dBc; the horizontal axis in MHz. The resolution of the display is fixed and follows the appropriate GSM specification.

The reference level, shown in the lower left corner of the display, is the power level measured at the center frequency in a 30 kHz bandwidth. All levels displayed in the live display are relative to the reference level (at the carrier frequency) and thus given in dBc. As a consequence, the ACPM modulation display will always show 0 dBc at the carrier frequency.

How to read the live display

The test limits according to the appropriate GSM specification for the single frequencies are shown in the live display by **horizontal red** lines.
When the measurement result for a single frequency is within the limits, the corresponding **vertical bar** will be displayed in blue (cyan for the measurements taken at ±100 kHz and ±250 kHz for easier reading).
A vertical bar displayed in **red** indicates that the current measurement result is beyond the limits set by the appropriate GSM specification (as it is for −1200 kHz, −1000 kHz, 1000 kHz and 1200 kHz in the example shown above).

Setting the center frequency

The center frequency of the spectrum analyzer is the frequency of the uplink carrier.
The center frequency cannot be modified in the RF spectrum analyzer menus.

– In synchronous mode (i.e. when a traffic channel has been set up), this frequency is given by the traffic channel number used and displayed on screen on the TCH display field. You can change the center frequency easily by entering a new channel number.
Example: TCH channel number 45 equals a center frequency of 889.0 MHz.

– In asynchronous mode (i.e. when there is no signaling exchange between the mobile under test and the 4400), the center frequency is identical to the frequency set on the analyzer frequency entry field of the Analyzer menus.

Averaging the ACPM modulation spectrum

The 4400 provides an additional tool for ACPM modulation spectrum measurements: the averaging function.

This means that a user-definable number of the continuous measurements will be taken and the single measurement results will be averaged. The result will be a smoother result curve, eliminating single peaks (that might be due to spikes or other measurement errors).

Working with the averaging tool

– Enter the averaging tool by pushing the |Start Avg.| softkey.

– The 4400 will now start the averaging process, using the number of measurements set on the Avg entry field.

– As long as the number of measurements taken is less than the number entered on the Avg entry field, the number of the current measurement will be displayed in red on the Count display field.

– As soon as the number of measurements taken is equal or larger than the number of measurements specified for the averaging process, the number on the Count display field will turn blue.

Important to note

– The entry range for the Avg entry field is from 1 to 500.

– The Avg entry field is not accessible unless the averaging process has been started.

– As soon as you change the current setting on the Avg entry field, the averaging process will be reset. This means that all previous measurements will be discharged and the series of measurements will be started anew.

– A reset of the averaging process can also be achieved by pushing the |Reset| softkey.

– When the number of measurements taken (display field Count) exceeds the number of measurements specified for the averaging process (Avg entry field), always the last Avg measurements will be averaged and displayed in the live display.

– In case the live display is frozen and unfrozen (using the appropriate softkey) during an averaging process, the averaging process will not be interrupted but will continue in the background.

– Even when the live display has been frozen, a reset of the averaging is possible, pushing the |Reset| softkey.

ACPM transient spectrum measurements

The mobile under test sends bursts with fast rising and falling edges. These switching transients cause interference in the adjacent channels. In order to minimize that interference, the mobile has to be compliant with the appropriate GSM specifications.

To call the menu, simply push the [ACPM trans.] marker tab.
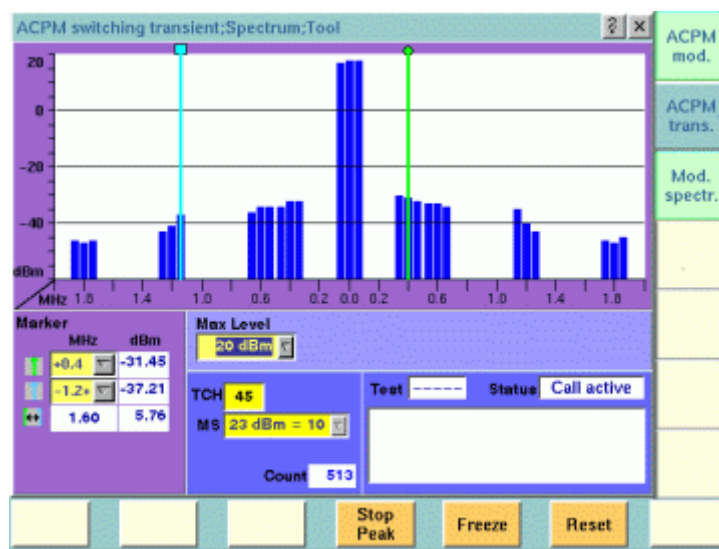
On first sight

The mobile's transient spectrum is displayed graphically in the menu's live display for the discrete frequencies specified by the appropriate GSM specification. The live display is updated continuously. To read the exact measurement at any position, simply use the display markers.
The current main "Parameters and status" are displayed in the lower right corner of the menu.
The softkeys provide access to important tools of the spectrum analyzer.

A glance into the GSM specifications

The ACPM measurements refer to section 4.2 (Output RF Spectrum) of specification GSM 11.10 (Mobile conformance specification; Radio transmission and reception).
Section 4.2.2 specifies measurements for the 'spectrum due to switching transients'.
These measurements are taken through a 30 kHz filter (resolution bandwidth; 100 kHz video bandwidth) at offsets of ±400 kHz, ±600 kHz, ±1200 kHz and ±1800 kHz from the carrier.
Measurements are ungated (i.e. measurements include the edges and the midamble).
The GSM specification then specifies that a single measurement shall be the search for the peaks of at least 10 bursts (peak-hold mode).

NOTE

The 'carrier frequency' is the frequency of the TCH used for this measurement.

NOTE

In addition to the requirements of the appropriate GSM specification, the RF power level measured at 30 kHz **above and below** the specified offsets from the carrier are displayed as well, in order to give additional indication when the measurement result for a single frequency is close to the limits.

NOTE

For clarity of results, the 18 additional measurement results not required by the appropriate GSM specification may be removed in the future.

Understanding the measurement principle

The measurement is performed in the frequency domain. Level measurements are performed at ±0 kHz, ±30 kHz, ±370 kHz, ±400 kHz, ±430 kHz, ±570 kHz, ±600 kHz, ±630 kHz, ±1170 kHz, ±1200 kHz, ±1230 kHz, ±1770 kHz, ±1800 kHz and ±1830 kHz offset from the nominal carrier frequency through a 30 kHz bandwidth filter. The measurement results are then displayed as vertical bars in dBm. If the Peak Hold function is activated, the results settle after about 10 measurements.

The spectrum live display

The live display is updated continuously (as long as it has not been frozen).
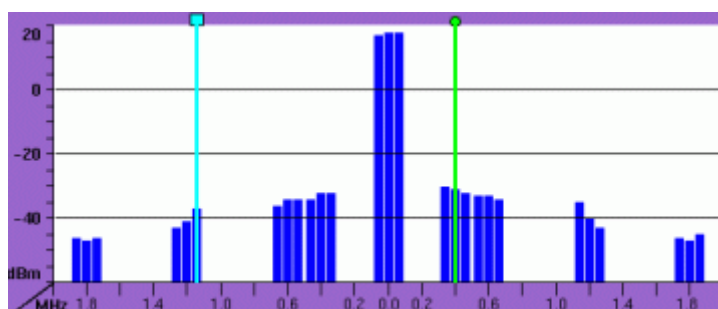
The vertical axis of the spectrum live display is in dBm; the horizontal axis in MHz.
The selection field `Max Level` allows to set the maximum power level to optimize the vertical resolution.
The horizontal resolution of the display is fixed and follows the appropriate GSM specification.

Entry field of the live display

Max Level — This selection field allows to set the RF power level represented by the top horizontal line of the live display.
Selections possible are: from +40 dBm to 0 dBm in steps of 10 dB.

NOTE

The value selected here should be equal or slightly higher than the expected maximum RF power level on the carrier frequency in order to avoid overload distortion and to achieve maximum vertical resolution.

Setting the center frequency

The center frequency of the spectrum analyzer is the frequency of the uplink carrier.
The center frequency cannot be modified in the RF spectrum analyzer menus.

– In synchronous mode (i.e. when a traffic channel has been set up), this frequency is given by the traffic channel number used and displayed on screen on the `TCH` display field.
Example: TCH channel number 45 equals a center frequency of 889.0 MHz.

– In asynchronous mode (i.e. when there is no signaling exchange between the mobile under test and the 4400), the center frequency is identical to the frequency set on the analyzer frequency entry field of the Analyzer menus.

Peak–hold function

As required in the GSM specifications, the measurements have to be carried out under peak-hold conditions. Therefore, the 4400 provides a user-selectable peak-hold function.
This means that the peak measurement result values out of a series of measurements will be searched for and displayed. The number of single measurement runs already taken into account is shown on the `Count` display field.
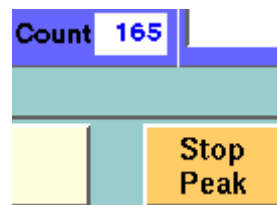
Working with the peak–hold function

– Start the peak-hold function by pushing the |`Start Peak`| softkey.

– The 4400 will now start the peak-hold process. This means that the peak measurement result value for every single (of the 27) measurement frequencies will be displayed on-screen. The number of single measurement runs taken into account is shown on the `Count` display field.
The softkey will change to |`Stop Peak`|.

Important to note

– A reset of the peak-hold process can be achieved by pushing the |`Reset`| softkey.

– In case the live display is frozen and unfrozen (using the appropriate softkey) during a peak-hold process, the peak-hold process will not be interrupted but will continue in the background.

– Even when the live display has been frozen, a reset of the peak-hold process is possible, pushing the |`Reset`| softkey.

Softkeys The softkeys of the RF spectrum measurement menus provide access to special functions and features.

**Softkeys of the modulation
spectrum menus**

Start Avg.	A push on this softkey starts the averaging tool of the spectrum analyzer. See description in section "Modulation spectrum measurements" or "ACPM modulation spectrum measurements" for reference. When the averaging tool is running, the softkey will change to |Stop Avg.|.
Stop Avg.	A push on this softkey stops the averaging tool of the spectrum analyzer and takes you back to the related modulation spectrum measurement. The softkey will change to |Start Avg.|.
Min-Max ON	Starts the concurrent display of minimum and maximum values (Modulation Spectrum only). This is like a peak-hold and a minimum-hold function; for each point on the horizontal axis, the smallest and the highest measurement values since the start of the test are shown together with the most recent measurement curve. The most recent measurement curve is shown in blue while the minimum and maximum curves are displayed in magenta, see picture below. Sampling minima and maxima can be restarted by pushing the **|Reset|** key. The minima and maxima display can be switched off by pushing the softkey again.
Min-Max OFF	This softkey is displayed when the minima/maxima display function is switched on (Modulation Spectrum only). It can be used to switch the minima and maxima display function off again.
Freeze	To freeze the current display for further or detailed analysis (or for printing), push this softkey. The softkey will change to |Unfreeze|. **Note:** After the |Freeze| softkey has been pushed, all measurements (and any ongoing averaging process) continue in the background, but the display is not updated. This means that even when the live display has been frozen, a reset of the averaging process is possible, pushing the |Reset| softkey.
Unfreeze	To unfreeze the current display, push this softkey. The softkey will change to |Freeze|.
Reset	A push on this softkey resets the averaging tool of the spectrum analyzer. See description in section "Modulation spectrum measurements" or "ACPM modulation spectrum measurements" for reference.

**Softkeys of the ACPM transient
spectrum menu**

Start Peak	A push on this softkey starts the peak-hold process of the "ACPM transient spectrum measurements". When the peak-hold process is running, the softkey will change to |Stop Peak|.

Stop Peak	A push on this softkey stops the peak-hold process of the "ACPM transient spectrum measurements" and takes you back to the continuous measurement. The softkey will change to |Start Peak|.
Freeze	To freeze the current display for further or detailed analysis (or for printing), push this softkey. The softkey will change to |Unfreeze|. **Note:** After the |Freeze| softkey has been pushed, all measurements (and any ongoing peak-hold process) continue in the background, but the display is not updated. This means that even when the live display has been frozen, a reset of the peak-hold process is possible, pushing the |Reset| softkey.
Unfreeze	To unfreeze the current display, push this softkey. The softkey will change to |Freeze|.
Reset	A push on this softkey resets the peak-hold process of the "ACPM transient spectrum measurements".

Minima and maxima display
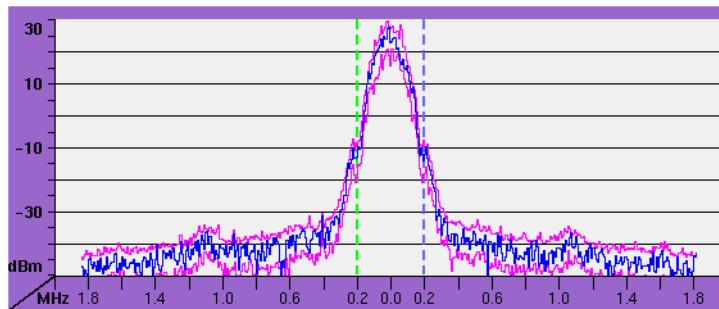
Display markers

The 4400 offers two markers for easy reading of the graphic spectrum displays:

Modulation spectrum measurement

On the kHz entry fields, you set the markers to the required horizontal positions (= offset from the carrier in kHz).
The corresponding result fields (dBm) show the measurement values obtained for the two positions in dBm.
The third row shows the difference between the selected frequencies and the difference between the two corresponding measurement values.

**ACPM modulation spectrum
measurement**

Marker MHz	dBc
+1.6	-74.43
-1.8	-72.64
3.40	1.79

On the MHz selection fields, you set the markers to the required horizontal positions (= offset from the carrier in MHz).
The corresponding result fields (dBc) show the measurement values obtained for the two positions relative to the RF power level of the carrier.
The third row shows the difference between the selected frequencies and the difference between the two corresponding measurement values.

**ACPM transient spectrum
measurement**

Marker MHz	dBm
+0.6	-77.51
0.0	-9.75
0.60	67.77

On the MHz selection fields, you set the markers to the required horizontal positions (= offset from the carrier in MHz).
The corresponding result fields (dBm) show the measurement values obtained for the two positions in dBm.
The third row shows the difference between the selected frequencies and the difference between the two corresponding measurement values.

NOTE

A selection of an offset frequency marked with a – behind (e.g. -0.6-) means the related offset frequency **minus 30 kHz**, while a + marker means the related offset frequency **plus 30 kHz**. A setting of e.g. -0.6- would then mean that the frequency to be measured at is –630 kHz from the carrier.

Parameters and status

On the spectrum analyzer menus, the following parameters and status information will be displayed:

NOTE

These parameters and status information is only relevant when working in synchronous mode (i.e. when a traffic channel has been set up). When measuring in asynchronous mode (i.e. without call setup), all display fields described here have no meaning at all.

Parameter display fields
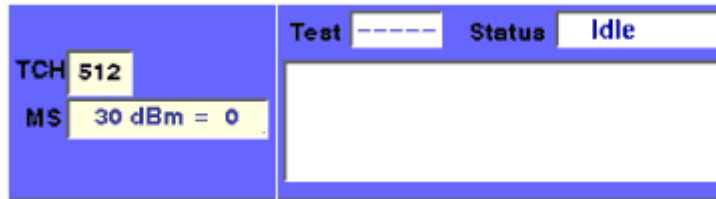
The meaning of the two display fields is as follows:

TCH — Here, the TCH channel number (in synchronous mode) is displayed.
This channel number reflects the carrier frequency (e.g. TCH 45 reflects an uplink carrier frequency of 889.0 MHz).
You can change the channel number to take measurements on a different frequency or band; the 4400 will perform a handover to the new frequency and restart measurements there.

MS — This display field shows the power class and the maximum RF power level of the mobile under test (in synchronous mode only).
You can also change the commanded power level by selecting another power level step.

Status information display

The status area shows the overall results of the current test and the signaling status.

Test

PASS — All measurement results of the current test are within their limits.

FAIL — This mark indicates that test limits have been violated by the measurement results.

---- (four hyphens) — Will be displayed when the measurement has not been started yet or when there are no limits available for the current test.

Status

Signaling status from the 4400's point of view. The large display field below gives more details and a short signaling history.

Idle — This is the status as long as there is no signaling active.

Call active — Indicates that there has been a call established.

Paging — This will be displayed as long as the 4400 pages the mobile.

Closed Loop — Will be displayed as long as there is a closed loop established on the mobile's side for e.g. the BER test.

– – – – – – (six hyphens) — This indicates that the call setup hasn't been started yet or that there has been no communication between the 4400 and the mobile so far.

Audio measurements

The Audio Option provides an AF generator and an AF analyzer.
The audio analyzer may also be used to perform measurements on audio signals transmitted over the air interface and converted back into analog signals by the optional codecs.

The coder part of an audio codec (coder and decoder) translates the analog signal or its binary representation into a compressed digital signal which can be transmitted on a traffic channel. The decoder part takes the compressed data from a TCH and transforms them to an analog signal or its binary representation. – An audio codec is sometimes also referred to as a **vocoder** (voice coder).

This section consists of the following parts:

– "Audio measurements – test basics" – This section gives an overview of the options needed for audio measurements and the basic test functionality.
– "Audio frequency generator" – Here you will find all information regarding the audio generator of the 4400 and the relevant parameters.
– "AF basic analyzer" – The so-called basic analyzer incorporates all nongraphic measurements (like AC voltage, DC voltage, frequency, distortion and SINAD).
– "AF spectrum analyzer" – This section will explain the built-in audio spectrum analyzer functionality of the 4400.

Audio measurements – test basics

In its basic configuration, the 4400 does not support audio measurements. In order to provide audio test functions, you need to install the Audio Option. This option provides your 4400 with both an audio generator and an audio analyzer.

To convert the digital data exchanged between the mobile under test and the 4400 into audible sound, you also need to add at least the Basic Codec Option. As you know, GSM currently uses three different codecs:

– **Full Rate** is what GSM started with and what is still the most common codec type. To convert Full Rate into audible sound, the Basic Codec Option is all you need.
– **Half Rate** is supported by most of the mobiles available. However, the network operators are reluctant to use this feature as they are mainly afraid of loosing speech quality.
– **Enhanced Full Rate** is a new Codec algorithm that will provide better speech quality than Full Rate. To convert Enhanced Full Rate into audible sound, you will need the Codec Extension Option in addition to the Basic Codec Option.

Input and output ports for audio measurements

The following connectors (interfaces) are used by the audio option:

– "AF in", a BNC connector on the front panel of the 4400. This connector is where the audio signal to be measured needs to be connected.
This input can be used either as a balanced or as an unbalanced input (soft-

ware switch).
The input impedance is approx. 250 kΩ||20 pF.
The usable AC frequency range stretches from 30 Hz to 20 kHz.

> **HAZARD**
>
> The **absolute maximum input voltage is ±40 V DC or 30 V (rms) AC.** An AF input voltage of more than ±40 V DC or 30 V (rms) AC may result in an **immediate destruction** of the highly sensitive AF input stage of the 4400. Willtek will not accept liability for any damage of the input stage due to overload.

– "AUX 4 (AUX in)", pins 12 and 13 of the AUX4 connector on the rear panel of the 4400. Audio signals of low voltage (as e.g. generated by a microphone) can be connected here.
  This input is **un**balanced.
  The input impedance is < 1 kΩ.
  The usable AC frequency range stretches from 30 Hz to 20 kHz.

> **HAZARD**
>
> The **absolute maximum input voltage is 1 V (rms) AC.** An AF input voltage of more than 1 V (rms) AC may result in an **immediate destruction** of the highly sensitive AUX in input stage of the 4400. Willtek will not accept any liability for any damage of the input stage due to overload.

– "AF out", a BNC connector on the front panel of the 4400. The audio signals generated or decoded by the 4400 can be made available at this connector. This output is **un**balanced.
  Best performance is achieved on a 600 Ω load.
  The maximum output voltages are 4 V (rms) for sinusoidal signals or 11 V peak-to-peak for all other signals.
– "AUX 4 (AUX out)", pins 8 and 9 of the AUX4 connector on the rear panel of the 4400. The audio signals generated or decoded by the 4400 can be made available at this connector.
  The output impedance is 8 Ω; the maximum output power level is 1 W.

> **HAZARD**
>
> The AUX out is a balanced output. This means that the voltages provided are **floating**.
> Never short-circuit these two pins and never connect them to ground as this could result in an **immediate destruction** of the loudspeaker output stage of the 4400.

– Last but not least, the 4400 provides the speech codec (vocoder) options as an additional input and output ports. The generated audio signal is converted to a binary coded format, which is modulated onto the RF carrier and transmitted over the "RF in/out" connector on the front panel. Similarly, the RF signal from the mobile can be demodulated, decoded and analyzed by the audio analyzer.

> **HAZARD**
>
> NEVER connect an audio input or output to the RF connector directly!

As detailed below, the 4400 provides several interfaces to connect to the mobile phone. But where do I find the appropriate connector on the mobile?

There are no standardized audio connectors on the mobile phones; each manufacturer provides proprietary connectors. These can be found typically on the car kit interface or on a test adapter. Please refer to the manufacturer for more details.

Access to the audio menus

There is access to the audio menus in two ways:

– For tests in asynchronous mode, push the **TOOLS** function key from the Welcome menu. If the audio option is installed, the tool bar provides access to the audio menus with a push on the [Audio] marker tab.

– For tests on a voice channel, select the appropriate system option and then push the |Calls & Meas.| softkey from the Welcome menu. Now, originate a call (either from the mobile under test or from the base station). When the call is active, push the **TOOLS** function key. If the audio option is installed, the tool bar provides access to the audio menus with a push on the [Audio] marker tab.

Codec

A voice codec is used to transform an audio signal into a bit stream that is then transmitted over the RF channel. There are two optional codecs available.
Instead of measuring an analog signal from the AF input or AUX in input, the digital audio signal transmitted over the RF traffic channel can also be analyzed. The voice coded option is required to translate the binary coded information into an 8 kHz sampled digital signal.
To the same token, the audio signal generated by the 4400 can be applied to the traffic channel via the optional voice codec. The RF signal can then be used to stimulate the AF parts of the mobile under test.
The codec algorithm itself (on the user interface of the 4400 often referred to as TCH Type) can be selected from all GSM test menus before the call is set up.

Important to note

– When no codec option is installed on the 4400, the appropriate codec algorithm is not available and the AF analyzer/generator will not work in the mode explained above, although it is possible to select `FullRate` or `Enh.FullRate` on all GSM test menus.

– In asynchronous mode, the `Codec out` or `Codec in` settings are not available on the corresponding menus, since the use of the codecs requires a traffic channel to be set up.

– It is not possible to change the codec algorithm during an ongoing call.

Audio frequency generator

This menu allows for the settings of the AF generator.

Using the AF generator

The AF generator can be set to generate audio signals from 20 Hz to 20 kHz at a defined output level and a selectable waveform. As an alternative to an internally generated single tone, the 4400 can also replay a recorded multitone signal which can be used to generate more complex signals and to analyze the audio parts on multiple frequencies at the same time.

The generated signal can be fed either to the analog outputs ("AF out" or "AUX 4 (AUX out)") or to the mobile using a previously established traffic channel. The typical purpose of an audio signal is to stimulate and check the AF circuitry of the mobile under test. The following test setups are used for that purpose:

- Testing the mobile's AF input filters, A/D converter and voice coder. In this case, the audio signals generated by the 4400 can be used and fed to the mobile using a headset input or an artificial mouth.
- Testing the mobile's voice decoder, D/A converter and AF output filter. In this case, the audio signals generated are sent to the mobile under test over the air interface and analyzed with the 4400's built-in Basic Audio Analyzer or the built-in Audio Spectrum Analyzer.

Main parameters of the AF generator

The main parameters that affect the setting of the audio generator can be found on the AF Generator menu. To gain access to this menu, push the [Gen.] marker tab.
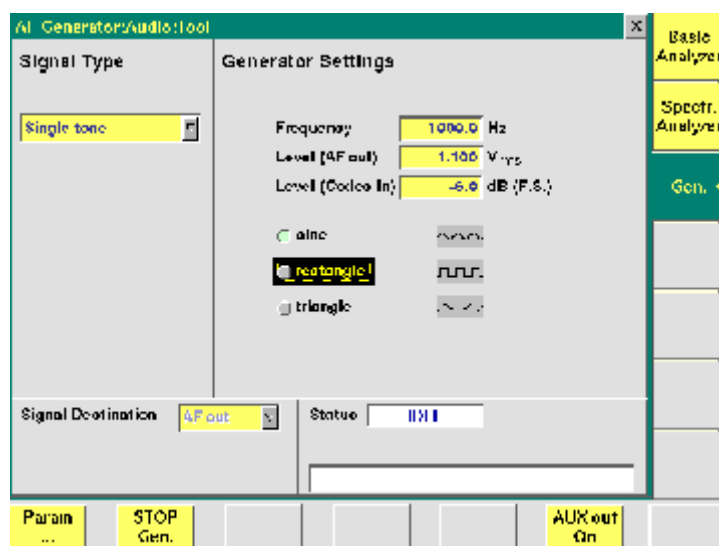
This menu consists of four areas:

- `Signal Type` allows to choose between `Single tone` and `WAV file` generation. See section below for more details on wave file replay.
- `Generator Settings` is the area where you may select frequency, output level and the waveform of the audio signal to be generated. This section of the menu is described below.

– `Signal Destination` provides a selection of where the generated audio signal shall be sent to. This field is accessible only for system options providing a voice codec (vocoder) option.

– `Status` gives an overview of the current signaling status while working in call mode.

AF generator settings

This area of the menu gives access to the main parameters of the audio generator.
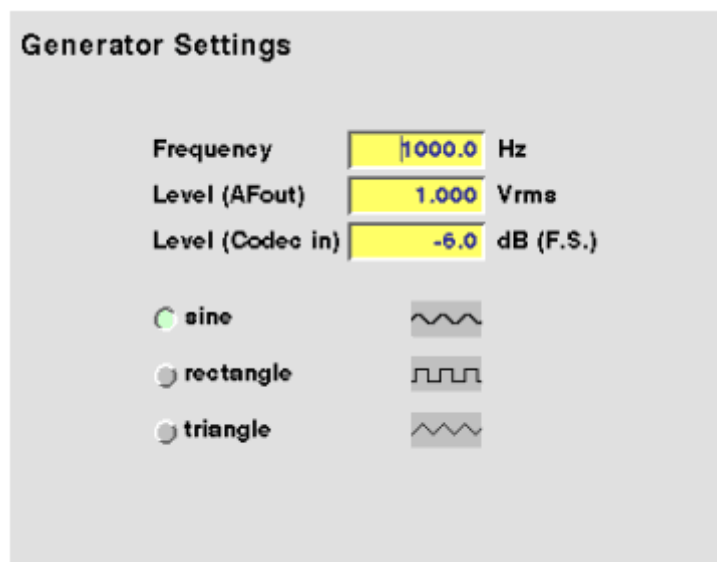
Frequency — Entry field to set the frequency of the audio generator.
Entry range: 20 Hz to 20 kHz in increments of 0.1 Hz.

NOTE

When using audio signals for transmission over the air interface, the audio frequency should be set in a range between 300 to 3200 Hz as any voice codec will only operate in this frequency range (just like a normal telephone line, where the audio signal transmitted is also limited to this frequency range).

Level (AF out) — Entry field to set the RMS-valued output voltage of the AF signal on the "AF out" connector.
Entry range: 0.0 V(rms) to 5.599 V(rms) in increments of 1 mV.

Level (Codec in) — Entry field to set the volume of the audio signal, when it is transmitted over the air interface and therefore connected internally to the input of the 4400's codec.
Entry range: –55.0 dB(F.S.) to 0.0 dB(F.S.) in increments of 0.1 dB.

NOTE

The physical unit dB relative to full scale (dB(F.S.)) means a level relative to the nominal maximum input level at the voice codec.

---

sine — Check box. If ticked, the output signal will have a sinusoidal waveform.

> **NOTE**
>
> When `Codec in` is selected as the `Signal Destination`, only sinusoidal signals will be generated, as the usable frequency range of all codecs is limited to 3200 Hz. As a consequence, any other waveform would be high-pass filtered and converted to some kind of sinusoidal waveform anyway.

rectangle — Check box. If ticked, the output signal will have a rectangular waveform.

triangle — Check box. If ticked, the output signal will have a triangular waveform.

AF signal destination
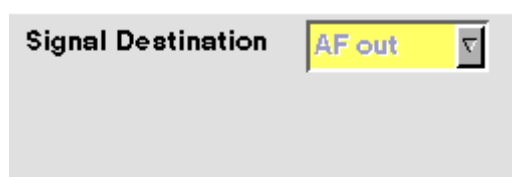
Signal Destination — Selection field. Selects where the audio signal generated is sent to.
Choices: `AF out` and `Codec in`.

> **NOTE**
>
> `Codec in` cannot be signal destination and AF analyzer source at the same time. If the selection field does not allow to select `Codec in`, first de-select `Codec out` as the signal source of the Basic Audio Analyzer or the Audio Spectrum Analyzer.

> **NOTE**
>
> `Codec in` cannot be selected unless a call setup has been performed.

> **NOTE**
>
> The audio signal can additionally be fed to the "AF out" connector. See section "Main parameters of the AF generator" on page 234 for details.

Wave file replay **Basics**

A single tone is useful to evaluate the signal quality in terms of voltage level, distortion and harmonics.

Multiple tones or a complex signal with varying frequencies and amplitudes can be useful to check the level over the entire audio spectrum in one go. It is also useful to evaluate the quality when testing over the air interface when Enhanced Full Rate Speech is selected; this codec algorithm is not suitable for sine wave tones as the audio level will vary.

The 4400 supports multitone or complex audio signals with a file replay function: A previously recorded audio signal can be stored on and replayed from the hard disk of the 4400; the signal must be stored as a *.WAV waveform file known from personal computers. See below for the specification of supported waveform files.

Complex audio signals are typically defined by an R&D lab for use by manufacturing and service, and can be generated with lab equipment such as an audio signal generator. The generator may be able to store the signal in a file; alternatively, the file can be created using the procedure described below. Example wave files are delivered with the 4400.

Creating a wave file using a signal generator and a PC

1   Set up the audio signal generator to provide the wanted signal to an analog output port.

2   Connect the analog output port of the signal generator to the audio input port of a personal computer.

3   Under Windows, start up the Audio Recorder in `Start > Programs > Accessories > Multimedia`.

4   Under `File > Properties`, select an appropriate PCM recording format; check the allowable formats in the Wave File Specifications section below. Remember Nyquist's theorem: The sampling frequency should be more than twice the maximum audio frequency!

5   Record the signal; ensure that the signal is recorded at sufficient voltage level without being distorted.

6   Save the recorded signal in a file on the PC's hard disk.

Copying the wave file onto the 4400

In order to load and replay the wave file, it should be available on the hard disk of the 4400. The following description tells you how to copy a file from a PC to the correct location on the 4400's hard disk.

1   Copy the file from your PC's hard disk to a floppy disk.

2   Insert the floppy disk in the floppy disk drive of the 4400.

3   In the Welcome menu, select `Tools > RAPID!`.

4   In the RAPID! file menu, select the file on floppy disk (/io/floppy/`file-name`.wav).

5   Select **ETC > Copy** to copy the file from the floppy disk to the /rapid/wave directory.

NOTE

In order to copy files from the floppy disk, the disk must be DOS-formatted and the file name must conform to the 8.3 format.

NOTE

You can use the TCP/IP Option and Willtek's Ethernet PC Card to copy wave files to the 4400 via a local area network (LAN). These files need not conform to the 8.3 file name convention.

Wave file specifications

Wave files can hold various sampling rates and data widths. The 4400 supports the following formats:

– Stored audio samples can be **8 or 16 bits** wide.

– For output to the AF out connector, the data (sampling) rate must be **48 kHz**.
For output to the audio codec, the sampling rate must be **8 kHz**.

– The signal should be recorded in **mono**.

– The maximum wave file size is 1,000,000 bytes.

Example: A 16 bit wide audio signal, sampled at a rate of 48 kHz, takes 96 kbytes per second, which means you can record 10 s for replay by the 4400. If a longer signal evaluation is required, the wave file can be replayed. Simply enable the `Auto-Repeat` field for the 4400 to play the wave file in a loop.

Signal type settings for wave file replay

If `WAV file` replay is selected, additional fields appear in the Signal Type section of the Audio Generator display. These fields allow you to set up the wave file replay options and display information about the selected file.
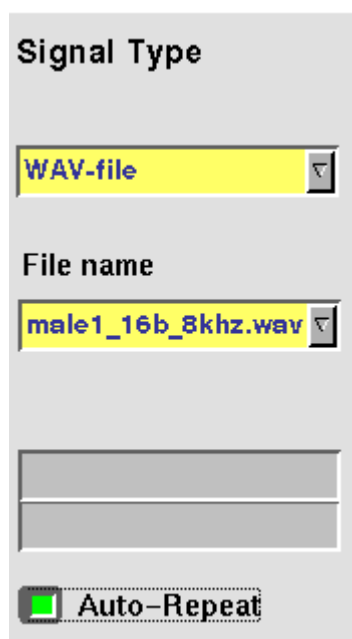
Signal Type — This input field allows you to select between the single-tone generator and the wave file replay.
Choices: `Single tone`, `WAV file`

File name — This field allows you to select a waveform file from the /rapid/wave directory.

NOTE

This field is visible only when wave file replay has been selected as the signal type.

Auto–Repeat — Check box. If ticked, the signal from the file is repeated over and over again until the generator is stopped with the STOP Gen. softkey.

AF generator menu – softkeys   The softkeys of the AF generator menu provide direct access to the following functions:

Provides direct access to the Parameters menu of the audio generator.

A push on this softkey stops the generation of audio signals and terminates all audio outputs.
The softkey will change to **START Gen.** Pushing this softkey will switch on the audio generator again with all previous settings and all outputs will be on the same connectors and levels as set before.

Pushing this softkey will turn on the output of the audio signal to the **AUX OUT** connector. All other settings of the audio generator will remain unchanged.
The softkey changes to **AUX Out Off**. A push on this softkey will turn the signal at the AUX out connector (for example to an external speaker) off again.

AF generator – parameters   The **PARAM...** softkey of the AF generator menu provides direct access to the AUX out controls:
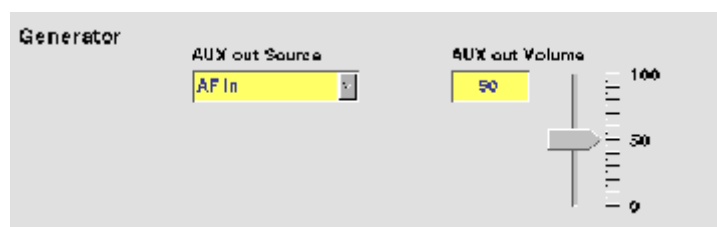
When, for example, an external loudspeaker is connected to the **AUX OUT** connector of the 4400, you can listen either to the audio signal generated or received by the 4400.
The AUX output can be activated or deactivated from any of the three Audio menus, using the |AUX out On| or the |AUX out Off| softkeys, respectively.

AUX out Source — Selection field. Selects the audio signal to be directed to the "AUX 4 (AUX out)" connector of the 4400.
Choices: AF in, Codec out and Audio generator.

━━━━   NOTE

Codec out cannot be selected unless a call setup has been performed.

▬▬ **NOTE**

`Codec out` cannot be AF analyzer source and signal destination at the same time. If the selection field does not allow to select `Codec out`, first deselect `Codec in` as the signal destination of the AF Generator.

AUX out Volume — Entry field to control the volume of the external speaker. Entry range: 0 to 100, where 0 means that no signal will be output while 100 stands for the maximum volume.

▬▬ **NOTE**

When an external mouse is connected to your test set, you may also set the AUX out volume by moving the volume slider up and down.

AF basic analyzer

The 4400 provides two different audio analyzers:

– The **basic** audio analyzer provides the most important numeric measurement results on one menu and also offers the possibility for statistic data evaluation.
– The "AF spectrum analyzer" provides a spectrum display of the audio signal.

The basic audio analyzer is described below.

The AF basic analyzer menu

The menu of the AF basic analyzer gives a quick overview of the **current** measurement results of the most important audio measurements. To gain access to this menu, push the [Basic Analyzer] marker tab.
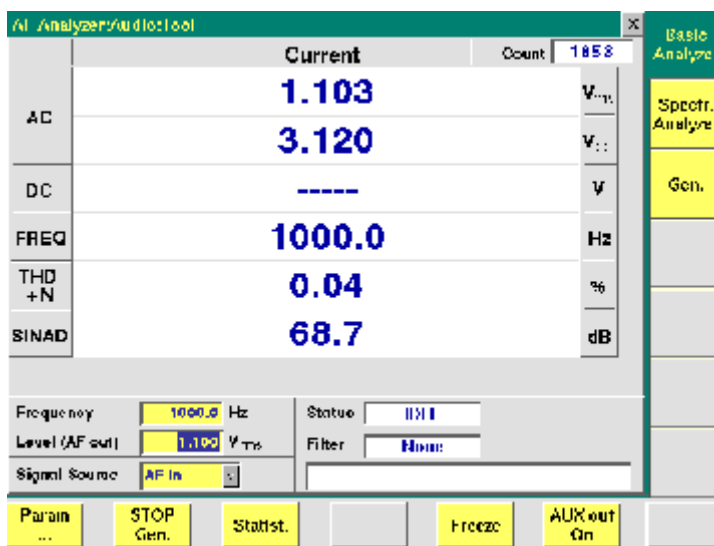
It consists of three areas:

– most of the menu is occupied by the **current measurement results** display,
– the **parameters** area provides access to the most important settings of the "Audio frequency generator", while

– the **status** area gives an overview of the current signaling status while working in call mode.
In addition, an audio filter selected will be displayed here.

The meaning of the softkeys is described in section Softkeys.

Current measurement results

This area of the AF Basic Analyzer menu provides the current results of the main audio measurements.

Count — Display field. Shows the number of the current measurement run since the counter was reset the last time.

AC — Display fields that give the level of the measured AC signal, either in V_{pp} (volts, measured peak-to-peak) and V_{rms} (volts, RMS-valued) or in dB(F.S.)(dB relative to full scale) for signals measured over the air interface.

DC — Display field. If DC coupling is selected, this display field shows the DC voltage component of the analog audio signal measured.

NOTE

This display field does not apply to measurements over the air interface, since the codec algorithms do not support DC components.

Freq. — Display field. In case the signal measured is a single-tone signal, the frequency of the signal is shown here in Hertz.

THD+N — This display field shows the distortion of a 1 kHz single-tone sine wave in percent (third harmonic distortion). THD+N stands for third harmonic distortion plus noise.

NOTE

No result will be shown when the signal measured is **not** a 1 kHz single-tone sine wave.

SINAD — This display field shows the signal-to-noise ratio, measured on a 1 kHz single-tone sine wave (expressed in dB). SINAD is an abbreviation for **Si**gnal to **N**oise **A**nd **D**istortion.

NOTE

No result will be shown when the signal measured is **not** a 1 kHz single-tone sine wave.

Parameters area

This area of the AF basic analyzer menu provides access to entry fields in order to set or change the most important parameters of the "Audio frequency generator".

Frequency	2000.0	Hz
Level (AFout)	0.050	Vrms
Signal Source	AF in	▽

Frequency — Entry field to set the frequency of the audio generator.
Entry range: 20 Hz to 20 kHz in increments of 0.1 Hz.

> **NOTE**
>
> When using audio signals for transmission over the air interface, the audio frequency should be set in a range between 300 to 3200 Hz as any voice codec will only operate in this frequency range (just like a normal telephone line, where the audio signal transmitted is also limited to this frequency range).

Level (AF out) — Entry field to set the RMS-valued output voltage of the AF signal on the "AF out" connector.
Entry range: 0.0 V(rms) to 5.599 V(rms) in increments of 1 mV.

Signal Source — This selection field allows to select the signal to be fed into the AF basic analyzer.
Selections possible are: `AF in`, `AUX in` and `Codec out`.
`AF in` means that the signal connected on the "AF in" socket will be analyzed, while
`AUX in` will analyze the signal connected to the "AUX 4 (AUX in)" connector.
Setting `Codec out` finally will forward the audio signal decoded by the 4400's internal codec to the AF basic analyzer.

> **NOTE**
>
> `Codec out` cannot be selected unless a call setup has been performed.

> **NOTE**
>
> `Codec out` cannot be AF analyzer source and signal destination at the same time. If the selection field does not allow to select `Codec out`, first deselect `Codec in` as the signal destination of the "Audio frequency generator".

AF analyzer – softkeys   The softkeys of the AF basic analyzer menu provide direct access to the following functions:
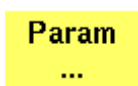
Param ...   Provides direct access to the Parameters menu of the AF basic analyzer.
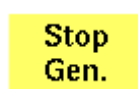
Stop Gen.   A push on this softkey stops the generation of audio signals and terminates all audio outputs.
The softkey will change to |`START Gen.`|. Pushing this softkey will switch on the audio generator again with all previous settings and all outputs will be on the same connectors and levels as set before.

Statist. | A push on this softkey provides access to the statistic data evaluation of the AF basic analyzer measurement results. The softkey changes to |Normal|. A push will take you back to the standard display of the AF basic analyzer.
Note: This softkey ist not available on the AF spectrum analyzer menu.

Reset | A push on this softkey resets the measurement run counter.
Note: This softkey is only available when the statistic data evaluation is on display.

Freeze | A push on this softkey freezes the display of the current measurement results. No further measurements will be taken. The softkey changes to |Unfreeze|. A push on this softkey will continue the measurements.

Aux out On | Pushing this softkey will turn on the output of the audio signals to the "AUX 4 (AUX out)" connector. All other settings of the AF basic analyzer will remain unchanged.
The softkey changes to |AUX out Off|. A push on this softkey will turn the external speaker off again with the settings as before.

AF analyzer – parameters

The **Param...** softkey of the AF analyzer menus provides direct access to the following settings:
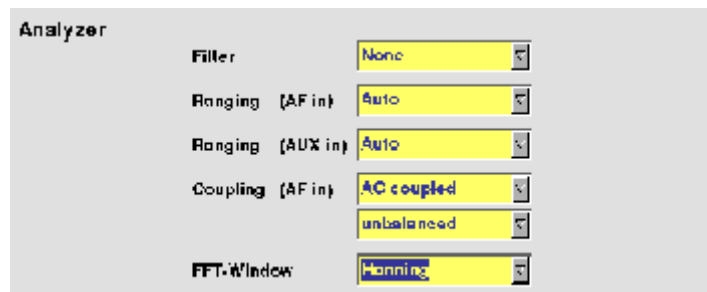
Filter — This selection field allows to set a weighting filter for the audio signals. The purpose of these filters is to fold the audio signals with the sensitivity curve of the human ear.
The settings possible are: None (default), CCITT and C-Message.
CCITT stands for a filter according to the (European) CCITT specifications, while C-Message means the American C-weighting filter.

Ranging (AF in) — Selection field. Used to set the expected level of the AC component of the audio signal connected to the "AF in" socket. The expected level is the peak-to-peak voltage of the AC component.
The settings possible are: Auto (default), 30 V, 3 V, 300 mV and 30 mV.

NOTE

For a quick availability of the first measurement result, avoid the AUTO setting as this involves a time-consuming adaptation process.

Ranging (AUX in) — Selection field. Used to set the expected level of the AC component of the audio signal connected to the "AUX 4 (AUX in)" socket. The expected level is the peak-to-peak voltage of the AC component.
The settings possible are: `Auto` (default), `1 V` and `100 mV`.

> **NOTE**
>
> For a quick availability of the first measurement result, avoid the `AUTO` setting as this involves a time-consuming adaptation process.

Coupling (AF in) — This selection field allows to switch the "AF in" connector from `AC coupled` (default) to `DC coupled` or to `DC coupled / LP`.

- If the input signal is a DC voltage only, select `DC coupled` to measure the DC voltage level.
- For a mixture of DC and AC signals, select `DC coupled / LP` to measure the DC voltage level and to suppress the AC components of the signal.
- For measurements of the AC component of the signal only, select `AC coupled` to achieve slightly higher accuracy.

> **NOTE**
>
> The settling time for the low-pass filter is approx. 300 ms. Therefore, the first measurements taken with the low-pass filter in place might be erroneous.

> **NOTE**
>
> Be sure to switch the low-pass filter off when calibrating the audio path.

Balancing — This selection field toggles the "AF in" connector between `unbalanced` (default) and `balanced`.

- `unbalanced` means there is one 'hot' signal line and one ground line. This is sometimes also referred to as 'asymmetric' or 'single ended'.
- `balanced` means that there are two 'hot' signal lines and no ground line (there may be a shield). This is sometimes also referred to as 'symmetric', 'differential', 'bridged' or 'floating'.

FFT-Window — Selection field. Allows to select the window for the **F**ast **F**ourier **T**ransformation of the audio signals.
The selections possible are: `Hanning`, `Hamming`, `Bartlett`, `Blackman/ Harris`, `Blackman`, `Rectangular` and `Flat Top`.

Statistical data evaluation   The AF basic analyzer also provides a menu for statistic data evaluation.
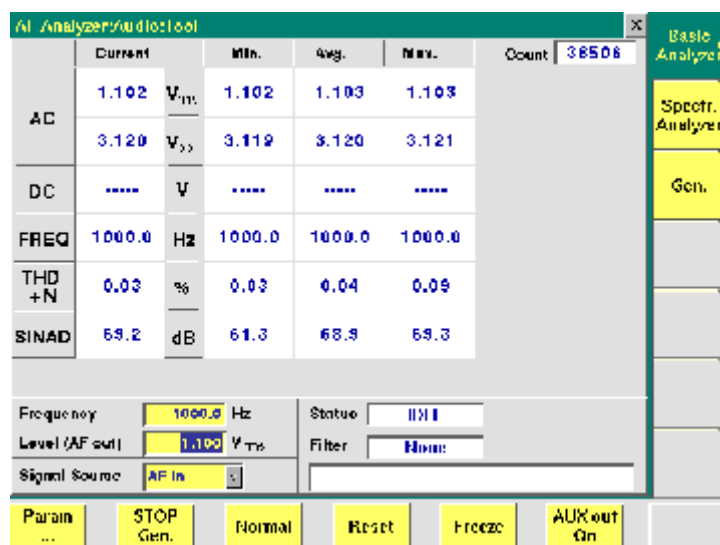
It consists of three areas:

– most of the menu is occupied by the **statistic data evaluation** display,

– the **parameters** area provides access to the most important settings of the "Audio frequency generator", while

– the **status** area gives an overview of the current signaling status while working in call mode.
In addition, an audio filter selected will be displayed here.

The meaning of the softkeys is described in section Softkeys.

The results area of the AF Basic Analyzer menu provides the current results of the main audio measurements.

Count — Display field. Shows the number of measurements taken into account for the statistic data evaluation. The counter can be reset using the **Reset** softkey.

Current — The display fields of this column show the measurement result values of the **current** measurement run.

Min. — The display fields of this column show the **minimum** measurement result values obtained during the past `Count` measurements.

Avg. — The display fields of this column show the **average** measurement result values obtained during the past `Count` measurements.

Max. — The display fields of this column show the maximum measurement result values obtained during the past `Count` measurements.

AC — The display fields of this line show the level of the AC signal measured, either in V_{pp} (volts, measured peak-to-peak) and V_{rms} (volts, RMS-valued) or in dB(F.S.) (dB relative to full scale) for signals measured over the air interface.

DC — The display fields of this line show the DC voltage component of the measured analog audio signal in case the input signal is `DC coupled`.

NOTE

This display field does not apply to measurements over the air interface, since the codec algorithms do not support DC components.

Freq. — The display fields of this line show the audio frequency in Hertz in case the signal measured is a single-tone signal.

THD+N — The display fields of this line show the distortion of a 1 kHz single-tone sine wave in percent (third harmonic distortion). THD+N stands for third harmonic distortion plus noise.

NOTE

No result will be shown when the signal measured is **not** a 1 kHz single-tone sine wave.

SINAD — The display fields of this line show the signal-to-noise ratio, measured on a 1 kHz single-tone sine wave (expressed in dB). SINAD is an abbreviation for **S**ignal to **N**oise **A**nd **D**istortion.

NOTE

No result will be shown when the signal measured is **not** a 1 kHz single-tone sine wave.

Parameters area This area of the menu provides access to entry fields in order to set or change the most important parameters of the "Audio frequency generator".

Frequency	2000.0	Hz
Level (AFout)	0.050	Vrms
Signal Source	AF in	▽

Frequency — Entry field to set the frequency of the audio generator.
Entry range: 20 Hz to 20 kHz in increments of 0.1 Hz.

NOTE

When using audio signals for transmission over the air interface, the audio frequency should be set in a range between 300 to 3200 Hz as any voice codec will only operate in this frequency range (just like a normal telephone line, where the audio signal transmitted is also limited to this frequency range).

Level (AF out) — Entry field to set the RMS-valued output voltage of the AF signal on the "AF out" connector.
Entry range: 0.0 V(rms) to 5.599 V(rms) in increments of 1 mV.

Signal Source — This selection field allows to select the signal to be fed into the AF basic analyzer.

Selections possible are: `AF in`, `AUX in` and `Codec out`.

`AF in` means that the signal connected on the "AF in" socket will be analyzed.

`AUX in` will analyze the signal connected to the "AUX 4 (AUX in)" connector.

The setting `Codec out` finally will forward the audio signal decoded by the 4400 internal codec to the AF basic analyzer.

NOTE

`Codec out` cannot be selected unless a call setup has been performed.

NOTE

`Codec out` cannot be AF analyzer source and signal destination at the same time. If the selection field does not allow to select `Codec out`, first de-select `Codec in` as the signal destination of the "Audio frequency generator".

AF spectrum analyzer   The AF spectrum analyzer is a useful tool for detecting spurious signals and harmonics. Signals from the following sources may be displayed on the spectrum analyzer:

– "AF in" connector,

– "AUX 4 (AUX in)" connector,

– `Codec out`, the output signal of the internal codec.

The AF spectrum analyzer menu   The menu of the AF spectrum analyzer provides a graphical display of the input signal's spectrum. To gain access to this menu, push the `[Spectr. Analyzer]` marker tab.
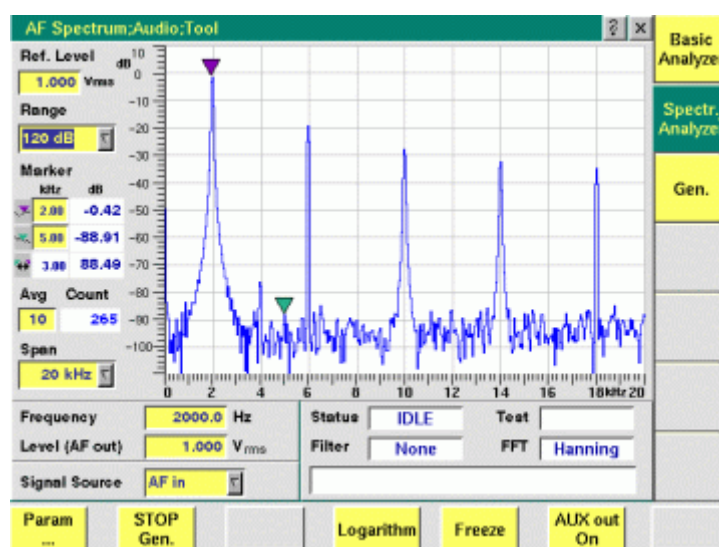
It consists of four areas:

– Most of the menu is occupied by the **spectrum display.**

–   The **display markers** make it quite easy to read the signal level for any frequency.

–   The **parameters** area provides access to the most important settings of the "Audio frequency generator".

–   The **status** area gives an overview of the current signaling status while working in call mode.
    In addition, the audio filter selected and the FFT Window chosen will be displayed here.

The meaning of the softkeys is described in section Softkeys.

The spectrum display

Ref. Level — The setting on this entry field determines the 0 dB line of the spectrum analyzer display.

–   When the signal source is either "AF in" or "AUX 4 (AUX in)", the reference level is expressed in V(rms).

–   When `Codec out` has been selected as the signal source, the reference level is identical with the full scale level of the codec (i.e. 0 dB(F.S.)).

Range — This selection field sets the resolution of the vertical axis. Settings possible are: `120 dB`, `80 dB` or `40 dB`.

> **NOTE**
>
> The top value of the spectrum display is always +10 dB.

Avg — This entry field allows you to initiate averaging of the last measurements. The number of measurements to be averaged can be entered here. Valid entries are in the range 1 to 500; an entry of 1 means that no averaging is performed.

Span — This selection field sets the displayed frequency span. Valid entries are **4 kHz** and **20 kHz**. The span defines the upper end of the displayed frequency axis; the display range starts at 0 Hz for the linear frequency axis and at 100 Hz for the logarithmic frequency axis.

Display markers

The AF spectrum display of the 4400 offers two markers for easy reading of the graphic display.

On the kHz entry fields, you set the markers to the required horizontal positions (= frequency). The corresponding result fields (*dB*) show the measurement values obtained for the two positions of the spectrum display.
The third row shows the difference between the selected frequency positions and the difference between the two corresponding measurement values.

Parameters area

This area of the AF Basic Analyzer menu provides access to entry fields in order to set or change the most important parameters of the "Audio frequency generator".

Frequency	2000.0 Hz
Level (AFout)	0.050 Vrms
Signal Source	AF in ▽

Frequency — Entry field to set the frequency of the audio generator.
Entry range: 20 Hz to 20 kHz in increments of 0.1 Hz.

> **NOTE**
>
> When using audio signals for transmission over the air interface, the audio frequency should be set in a range between 300 to 3200 Hz as any voice codec will only operate in this frequency range (just like a normal telephone line, where the audio signal transmitted is also limited to this frequency range).

Level (AF out) — Entry field to set the RMS-valued output voltage of the AF signal on the "AF out" connector.
Entry range: 0.0 V(rms) to 5.599 V(rms) in increments of 1 mV.

Signal Source — This selection field allows to select the signal to be fed into the AF basic analyzer.
Selections possible are: AF in, AUX in and Codec out.
AF in means that the signal connected on the "AF in" socket will be analyzed.
AUX in will analyze the signal connected to the "AUX 4 (AUX in)" connector.
The setting Codec out finally will forward the audio signal decoded by the 4400's internal codec to the AF basic analyzer.

> **NOTE**
>
> Codec out cannot be selected unless a call setup has been performed.

> **NOTE**
>
> Codec out cannot be AF analyzer source and signal destination at the same time. If the selection field does not allow to select Codec out, first de-select Codec in as the signal destination of the "Audio frequency generator".

HSCSD-multislot option

HSCSD (= High Speed Circuit Switched Data) enhances standard GSM with the bundled use of several time slots within a GSM frame for data transmission. Theoretically, the maximum data transmission rate in HSCSD is n * 14,400 bps, where n is the number of the bundled time slots. In case of the 4400, the maximum for n equals 4; this means that the 4400 can transmit and/or receive on four time slots per frame; each slot is set to carry 9.6 kbps.
On the 4400, the tool to perform tests on HSCSD mobiles is called "Multislot option".

This is what you will find in this section:

"Multislot – test basics" – What you should know before starting a test

"Tests of the HSCSD-Multislot option" – The specific tests of the Multislot environment

"System parameters" – Learn more about the specific system parameters of the Multislot environment

"Multislot-related SCPI commands" – The Multislot option incorporates a couple of specific SCPI remote control commands. This section will tell you all about those commands.

"Frequently asked questions" – This is where you will find detailed technical explanation and answers to common questions regarding the Multislot option.

Multislot – test basics

Introduction
The data transmission rate of the air interface in GSM is limited to 9.600 bps (or 14.400 bps with reduced data protection on the air interface). Due to the convergence of different IT applications (data transmission, fax, email), this tight bandwidth limitation is no longer acceptable. Many service providers and hardware manufacturers therefore turn towards the GPRS standard (= **G**eneral **P**acket **R**adio **S**ervice). GPRS uses dynamically allocated air time and provides a theoretical data rate of up to 176 kbps, but does not use a fixed channel allocation as in standard GSM.
For high-speed applications requiring dial-in lines, HSCSD (= High Speed Circuit Switched Data) offers the bundled use of several time slots within a GSM frame. Theoretically, the maximum data transmission rate in HSCSD is n * 14.400 bps, where n is the number of the bundled time slots. In case of the 4400, the maximum for n equals 4; this means that the 4400 can transmit respectively receive on four time slots per frame.

How multislot testing works on the 4400
The Multislot option of the 4400 allows testing the multislot transmitter in synchronous (or call) mode.

In synchronous mode, the mobile performs a full call setup. This means, it has to synchronize to the base station simulated by the 4400 and may then perform a call setup for an HSCSD data link.

The GSM tests of the Multislot option are very similar to the standard GSM tests like burst quality, phase and frequency error. To a large extent, the test menus of the Multislot option are identical with the standard GSM test menus.

Missing Pass and Fail indicators

The **Pass** or **Fail** indicators for the power/time template are not available in Multislot. The reason for that is the limited amount of space on the menus. As Willtek wanted to keep them as clear as possible, the Pass/Fail indication on Multislot has been left out.
All measurement results are, however, accessible using the SCPI commands and can thus be read out by an external controller.

Synchronous mode test setup

When the 4400 is in Idle mode (this means that there is no test and no signaling active), you may call the Multislot option from the GSM tool bar.

Performing tests in synchronous mode

– The mobile initially has to perform a GSM call setup (mobile originated) for a data call. Therefore, the 4400 will provide a base channel and simulate a base station. The 4400 will complete the call set up and guide the mobile onto an HSCSD link.

– As soon as this has been completed, the `Call active` message will be displayed in the status area of the corresponding test menu.

– You may then call any test menu. You may also vary parameters and display functions.

– As long as the call is active, the 4400 will display the two softkeys |Spectrum...| and |BS Release|.

– Changing the uplink or downlink configuration during a call is not possible. You can, however, release the ongoing call, change the configuration and set up a new data call.

– Call termination may either be initiated by the mobile or by pushing the 4400's |BS Release| softkey.

Tests of the HSCSD– Multislot option

As Multislot is an expansion to standard GSM, you will mainly perform the standard GSM tests like burst quality, phase and frequency error. Compared to the standard GSM test menus, there are just minor changes on the corresponding Multislot test menus. The following tests can be performed within the Multislot option:

– "HSCSD phase/frequency error test"
– "HSCSD burst test menu"
– "ACPM due to modulation test menu"
– "ACPM switching transient test menu"

HSCSD phase/frequency error test
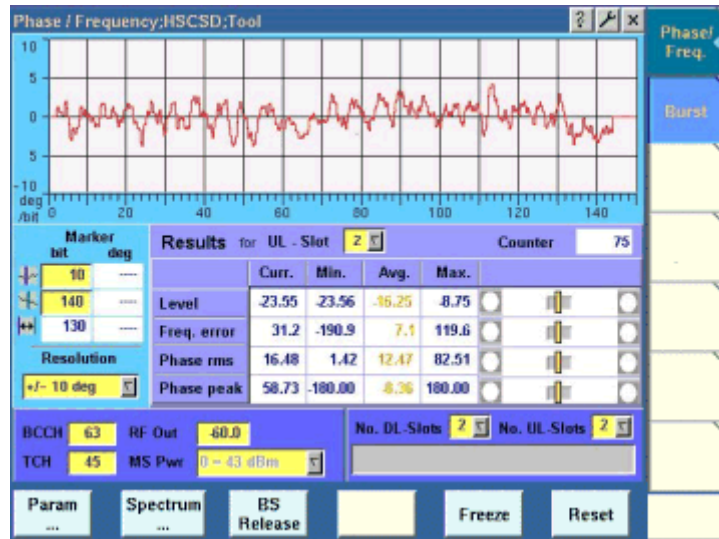
Main differences to the standard GSM Phase/Frequency error test menu

There may be up to four bursts per TDMA frame transmitted by the mobile. There-fore, you need to tell the 4400 which one of the mobile's bursts it shall evaluate and display graphically. There are also two additional entry fields where you configure how many DL time slots the 4400 is to transmit and how many UL time slots the mobile is going to transmit within a TDMA frame.

Results for UL–Slot — Here you enter the time slot that is to be evaluated. All graphics and measurement results are related to this time slot. Any change of the value will result in the current test being stopped and restarted automati-cally, using the new value.
Entry range: 1...4

NOTE

With the help of the related SCPI commands, you can, of course, read out the measurement results for all uplink (=UL) time slots of the same TDMA frame.

No. DL–Slots —
Here you set the number of downlink (=DL) time slots that the 4400 is going to transmit.
Entry range is 1...4.

No. UL–Slots — This entry field gives you the possibility to set the number of UL time slots that the 4400 shall receive (i.e. the number of slots per TDMA frame period that will be transmitted by the mobile).
Entry range is 1...4.

NOTE

The 4400 will be able to read out the mobile's Class Mark 3 from uplink mes-sages. This information element tells how many time slots within a TDMA frame the mobile is able to use for data transmission.
If your entry exceeds the mobile's capabilities, the 4400 will ignore the entered value and use the maximum number of time slots the mobile supports for transmission.

HSCSD burst test menu
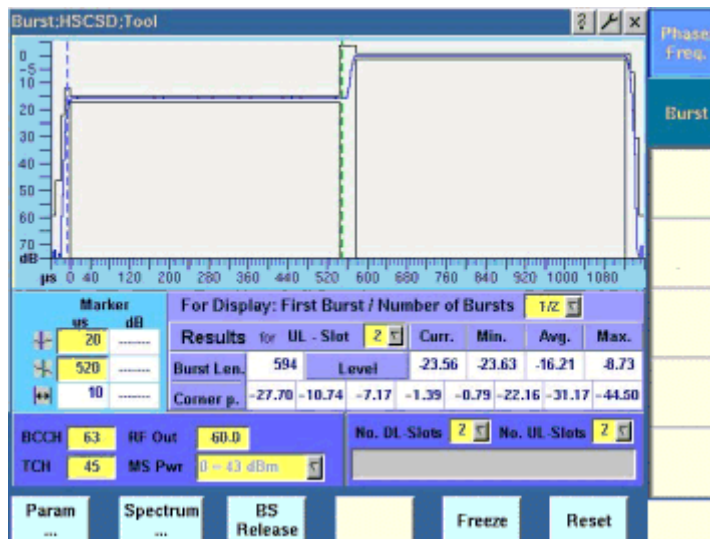
Main differences to the standard GSM Burst test menu

In Multislot, several bursts (up to 4) are displayed in this test menu. Thus, this menu not only helps you to evaluate the behavior of the mobile under test during and between the single bursts. It also provides you with a quick method to check if the mobile is able to vary its RF power level from burst to burst.

For Display: First Burst / Number of Bursts —

Here you select which burst(s) you would like to get displayed. The 4400 will always show you consecutive bursts. Therefore, it is sufficient to tell the 4400 the number of the first burst to be displayed and the total amount of bursts.

Example: Your mobile under test transmits two bursts. When you enter ½ on this entry field, the 4400 will display two time slots beginning from the first one it receives. Consequently, you will see the two bursts of the mobile being displayed in the display area.

NOTE

Any change on this entry field will result in the current test being stopped and restarted automatically, using the new parameter values.

NOTE

The display will only be altered or updated when a new test result has been received from the mobile.

NOTE

Please ensure that the selected number of time slots is in line with the two entry fields *No. DL-Slots* and *No. UL-Slots* of this menu.

> **NOTE**
>
> If you select time slots that are not being measured, the 4400 may not show any new measurements.

Results for UL-Slot — In order to achieve meaningful numeric test results and statistics, the 4400 does all the processing on the same time slot over many GSM frames.
On this entry field, you select that the time slot the 4400 is to do the calculations on.
Entry range is 1 to 4.

> **NOTE**
>
> With the help of the related SCPI commands, you can read out all test results of all uplink (=UL) time slots of the same frame.

> **NOTE**
>
> Please ensure that the selected UL slot is in line with the entry field *No. UL-Slots* of this menu.

> **NOTE**
>
> If you select a time slot that is not being measured, the 4400 may not show any new measurement results.

No. DL-Slots — No. DL-Slots `2` No. UL-Slots `2`
Here you set the number of downlink (=DL) time slots that the 4400 is going to transmit.
Entry range is 1...4.

No. UL-Slots — This entry field gives you the possibility to set the number of UL time slots that the 4400 shall receive (i.e. the number of slots that will be transmitted by the mobile per TDMA frame period).
Entry range is 1...4.

> **NOTE**
>
> The 4400 will be able to read out the mobile's Class Mark 3 from the uplink messages. This information element tells how many time slots within a TDMA frame the mobile is able to use for data transmission.
> If you enter a value on this entry field that is larger than the value transmitted by the mobile in Class Mark 3, the 4400 will ignore your entry and set it to the maximum number of time slots the mobile is able to use for transmission.

HSCSD spectrum test menus

There are three Spectrum test menus for the Multislot option as there are for standard GSM:

- The ACPM modulation menu shows you the mobile's adjacent channel power due to modulation.

- The ACPM switching transients menu displays the spectrum of the mobile's RF output spectrum due to switching transients.

– The Modulation Spectrum menu provides a quasi-continuous modulation spectrum display with selectable span.

Main differences to the standard GSM ACPM test menu

There may be up to four bursts being transmitted by the mobile within a TDMA frame period. Therefore, you have to tell the 4400 which one of the mobile's bursts it shall calculate the spectrum of and display it graphically.

ACPM due to modulation test menu
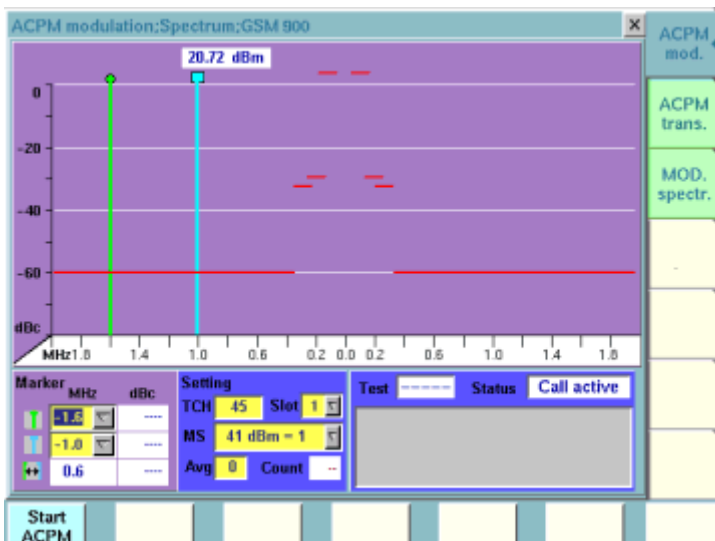
Slot —

On this entry field you select the time slot that the 4400 shall use for calculation and display of the spectrum.
Entry range: 1...4

NOTE

Any change of the value on this entry field will result in the current test being stopped and restarted automatically, using the new value.

ACPM switching transient test menu
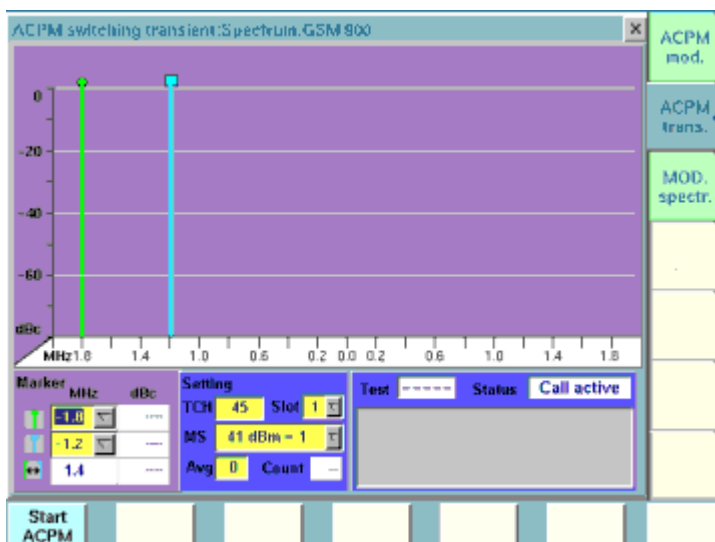
Slot —
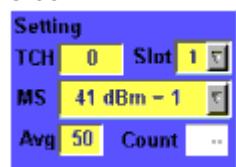
Here you set the number of the time slot you want the 4400 to display.
Entry range: 1...4

> **NOTE**
>
> Any change of the value on this entry field will result in the current test being stopped and restarted automatically, using the new value.

System parameters

Additional parameters for HSCSD
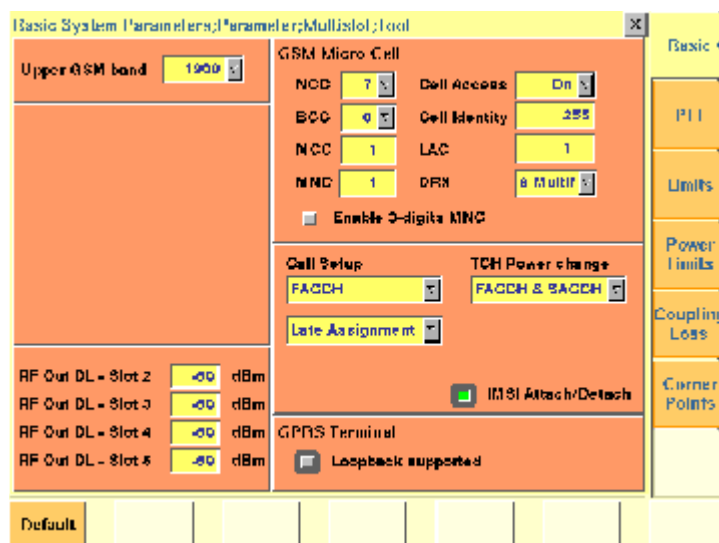
Main differences to the standard GSM Basic System Parameters menu:

The menu is arranged in a slightly different way. There are now four entry fields for the individual RF power levels in the lower left corner of this menu.

RF Out DL – Slot n **(n=2 to 5) —**

RF power level in dBm for the respective downlink slot (time slot 2 through 5). This is the power level that the 4400 emits on that slot.
Entry range: −120.0 dBm to −10.0 dBm

NOTE

When you are working with the Multislot option, the 4400's main TCH is set to time slot 4. This cannot be altered; therefore, the entry field *TCH Slot No.* is missing in the Basic system parameters menu of the Multislot option.

NOTE

The individual power levels are not yet supported. This means that you can enter values but the 4400 still transmits a continuous output power level.

Additional PTT parameters in HSCSD

Main differences to the standard GSM Power/Time Template (PTT) menu:

While there is just one PTT parameter menu for standard GSM, there are two PTT parameter menus for the Multislot option: the **PTT Normal menu** and the **PTT Trans menu**.

PTT Normal menu

As you know from the PTT in standard GSM, the values entered on the entry fields of the PTT Normal menu are the limits of a single burst.
While working with the Multislot option, the values entered on the PTT Normal menu are now the limits for the mobile's entire transmission within a TDMA frame. In other words: The upper and lower limits of the RF power level for the useful part of the burst are set on the PTT Normal menu. In Multislot operation, the limits set on this mask for the rising edge will be interpreted as the limits for the rising edge of the first burst sent by the mobile within a TDMA frame. Consequently, the limits set for the falling edge in the PTT Normal menu will be interpreted as the limits for the falling edge of the last transmitted burst within a TDMA frame.

NOTE

The entry field *Paging* is not available while working with the Multislot option.

PTT Trans menu

The template for the behavior of the mobile station between individual (consecutive) bursts of the same TDMA frame is set in the PTT Trans menu.
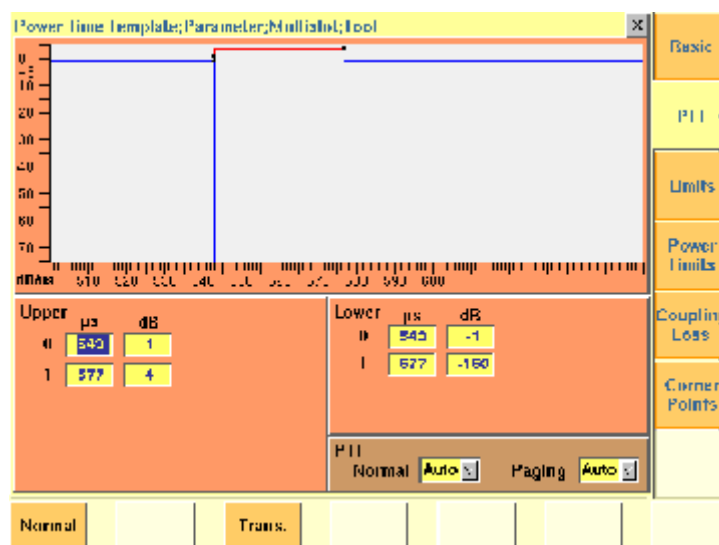
The values are set as on the PTT Normal menu.

NOTE

For every single burst, the average UL RF power level of the useful part of the burst is calculated by the 4400. This average UL RF power level is then used to position the user-defined template (upper and lower limits) on the display.

NOTE

In order to keep the test menus as clear as possible, no pass/fail indication will be displayed while working with the Multislot option. However, the corresponding measurement values can be read out any time using the related SCPI commands.

Multislot–related SCPI commands

On the 4400, there are so-called SCPI commands available for test set integration into your test environment (SCPI: Standard Commands for Programmable Instruments). With the help of the SCPI commands, you may change settings on the 4400 or read out measurement results.

Specifically for the Multislot option, the following eight commands have been added:

Function	Syntax
Switching Multislot option on/off	CONFigure:GSM:MSLot
Setting the number of DL time slots	CONFigure:GSM:BS:MSLot
Setting the number of UL time slots	CONFigure:GSM:MSTAtion:MSLot

Function	Syntax
Selecting the time slot to be measured on	`CONFigure:DIS-Play:GSM:MSLot:RESult`

Please refer to the SCPI chapter for full details.

Frequently asked questions

The Willtek 4400 seems to ignore my entries for the number of time slots to be used

While working in Sync, there is ongoing signaling between the mobile under test and the Willtek 4400. With its parameter **Transmit & Receive** in Class Mark 3, the mobile under test notifies the Willtek 4400 about the maximum number of time slots it can handle within a TDMA frame for an HSCSD data link.
If you enter a number of expected UL time slots that exceeds the capabilities of the mobile, the Willtek 4400 will ignore your entry, display an error message (in manual mode) and set the value to the maximum number of time slots the mobile is able to handle.

The call seems to be active, but I don't see any measurement results. What's going wrong?

The results can be presented for a selectable time slot. If a time slot has been selected that the mobile is not using, no results will be displayed. In order to see measurement results, select another (lower) slot number.

What RF power level will the Willtek 4400 use for DL ?

In the present software version, the 4400 will use the standard RF Out power level.

What does the Willtek 4400 transmit during the DL bursts ?

The Willtek 4400 allocates data traffic channels with 9.6 kbps. All DL bursts of one TDMA frame transmitted by the Willtek 4400 contain identical data.

RAPID!

<div style="text-align: right; font-size: 4em; font-weight: bold;">7</div>

This chapter describes how to use and how to program RAPID! applications. Main topics described in this chapter are as follows:

Overview

RAPID! is a combination of the simple-to-use programming language BASIC and the powerful SCPI command language, developed for the 4400.
RAPID! turns your 4400 into a fully automated radio test set that takes measurements, prompts the operators for inputs and generates a test result report that may be printed or sent to the manufacturing line control computer by LAN.

This chapter gives you all the information you need for successful editing of programs, for running and debugging them. It is divided into four subsections:

– Using RAPID! – Gives an introduction and explains everything about loading and saving files as well as editing, running and debugging programs.

– RAPID! syntax – This chapter is the 'Syntax manual' of RAPID!. It outlines the general syntax, the program line syntax, the use of variables, constants, operators and expressions.

– Commands – This part of the RAPID! reference contains a precise explanation of available commands together with application examples.

– Functions – A range of functions is available to work with numerical and string variables, as well as for input and output handling.

– Tables – Should any error occur during editing a program or during runtime, this subsection explains the reason for that in detail and gives information on how to trace back an error and to debug it.

NOTE

For all work within the RAPID! environment, we strongly recommend the use of an external keyboard (PS/2 connector) and a standard PC mouse (Sub-D 9-pin connector). For more details, please check with section "Connectors on the rear panel" on page 10.

Using RAPID!

This subsection gives an introduction and explains everything about loading and saving files as well as editing, running and debugging programs.

– Introduction – Here you will find a general explanation of what RAPID! is, and how to enter the RAPID! working environment.

– File menu – In this menu, you load, save, copy, rename or erase program files.

– Edit menu – The RAPID! editor provides powerful tools for entering code or for searching and replacing contents.

– Run menu – Here you select the file to run and start a program.

– Debug menu – This powerful menu provides you with all the features you need for successful program development: running a program in single steps, watching and altering variables and more.

Introduction

RAPID! is a programming language that turns the 4400 into a fully automated radio test set. Controlled by RAPID!, the 4400 will for instance execute a series of tests with minimal human interaction, and at the same time produce a test report with a clear evaluation of the results measured.

Of course, you may halt any program and show e.g. adjustment instructions on the display of the 4400. Once the operator responds to those instructions, the RAPID! program will continue with new measurements and/or instructions.

RAPID! is a great time saver for extensive tests that are performed frequently. Typical examples are automated acceptance tests after manufacturing or regular routine checks as part of maintenance or quality assurance.

RAPID! = BASIC + SCPI

RAPID! programs are a powerful combination of the simple-to-use BASIC programming language and the 4400 specific SCPI commands. The BASIC instructions implemented are outlined in detail in subsection RAPID! syntax while Chapter 8 is fully dedicated to SCPI.

BASIC commands

BASIC (**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode) commands provide the required program sequence. They also enable the further processing of measured results, the entry of numeric values and texts (character strings) as well as the formatted output of reports to e.g. a printer.

SCPI commands

SCPI commands are used for configuring the 4400 and for executing and evaluating measurements. You'll find more about the use of SCPI commands in subsection "SCPI and RAPID!" on page 347.

Entering and exiting the RAPID! environment

How to enter

To enter the RAPID! environment, navigate to the Welcome menu. There, simply push **TOOLS** > [RAPID!].

If an AUTOSTART.bas programm has been loaded and is active, the 4400 will now start that program. Otherwise the 4400 will now display the RAPID! File menu.

With the help of the File menu, you may select the file you want to edit or create a new file with the help of the related softkeys and marker tabs.

How to exit

The [Exit] marker tab quits the RAPID! environment and takes you back to the Welcome menu.

In case you did any program editing without saving the changes, the 4400 will prompt you to either save or ignore the previous editing:

You may then save the changes by pushing the **Yes** softkey, while a push on softkey **No** will discard any changes and leave the file as it was. **Cancel** takes you back to the RAPID! environment.

Marker tabs of the RAPID! environment

The marker tabs offer direct access to the menus of the RAPID! environment:

File menu – Loading, saving, copying, renaming and erasing files in general.

Edit menu – Selecting a file to edit, entering code, saving a single program file, searching for and replacing text.

Run menu – Selecting a file to run and starting a program.

Debug menu – Running a program, single-stepping a program, watching and altering variables.

Quitting the RAPID! environment.

File menu   In a similar way as e.g. Windows Explorer, the File menu gives access to the directory structure of the internal hard disk drive and allows to search for and select files.

> **NOTE**
>
> Some 'typical Windows' functions, e.g. double-clicking on a file name to open it, are not included.
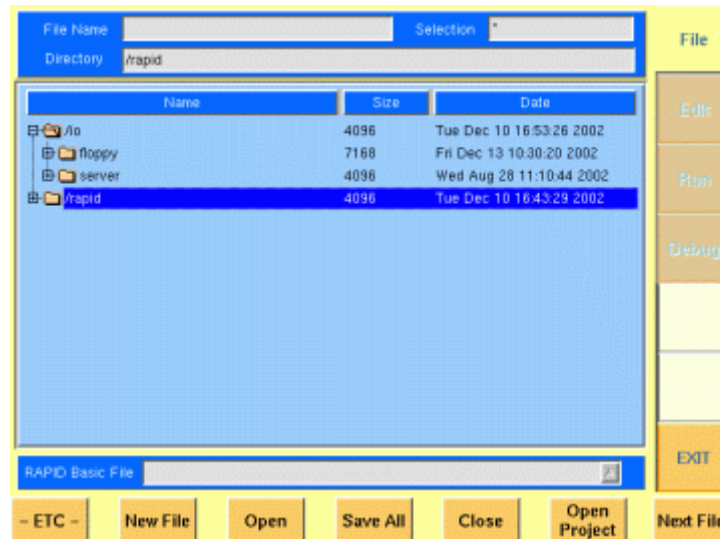
To keep work with the File menu as simple as possible, it is subdivided into three main sections:

– File name – directory – selection area, where you may select directories, file names and filters for the display of file names in the browser area.

– Browser area – this is where the 4400 shows its directory structure and allows you to browse through all directories and files.

– RAPID! basic file area – this is a selection field where you may select one of the previously opened RAPID! Basic Files.

The softkeys are used to call specific file functions (as e.g. creating a new file, opening a file or deleting a file) while the marker tabs give access to the various menus of the RAPID! environment.

> **NOTE**
>
> As long as no file has been opened or newly created, all marker tabs (except [File]) are grayed out and cannot be selected.

File name – directory – selection area   In this area, you may manually select a directory, a file name and/or the filter for the files displayed in the browser area.
To gain access to these three entry fields, simply click on the related entry field with the mouse or push the |New File| softkey in the file menu.

To navigate from one entry field to the subsequent one, simply use the **PGUP** and **PGDN** keys.

File Name — This is where you enter the name of the file (including extension). In case you enter an invalid file name, the 4400 will display a warning message on-screen.
To confirm the file name and its directory, simply push the **ENTER** key. When creating a new file, it will be added to the RAPID Basic File list automatically.

NOTE

The easiest way to enter a file name is of course with the help of the external keyboard. If it is currently not available, all characters are accessible through the Typing tabs.

`Directory` **—** Here, the 4400 displays the currently selected directory. In order to save a newly created file in a different directory or to change the currently selected directory, just enter the name of the directory you want to work within.

NOTE
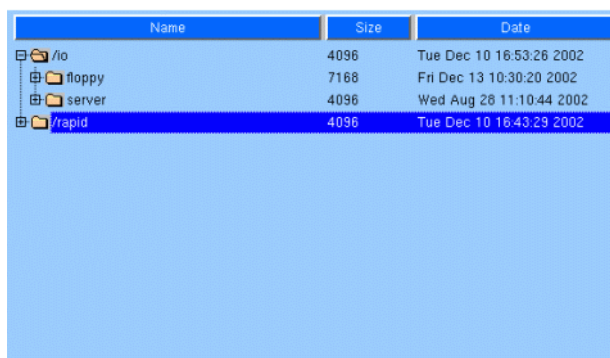
The directory path displayed here is the same as selected in the browser area, as long as you do not change it.

Selection — This entry field sets a filter for the file display within the browser area. The default setting is * and will make the browser display all files (with any extension).

Browser area   In this area, you may browse through directories just as with any other standard browser.
As soon as you enter the RAPID! environment, the file menu is automatically displayed and the browser area is active.

Name	Size	Date
/io	4096	Tue Dec 10 16:53:26 2002
floppy	7168	Fri Dec 13 10:30:20 2002
server	4096	Wed Aug 28 11:10:44 2002
/rapid	4096	Tue Dec 10 16:43:29 2002

The most convenient way to navigate through the Browser area is with the help of the mouse. Alternatively, use the **PGUP** and **PGDN** keys to scroll by pages and the **UP** and **DOWN** keys to scroll by lines.

A directory is opened or closed by a simple push on the **ENTER** key after it has been selected.

To open a file, mark it with the cursor (it is then highlighted) and push the **Open** softkey. As soon as you open a RAPID! file, it is automatically added to the RAPID Basic File list.

The top level directories are:

- **/io**. This directory includes the **floppy** subdirectory which in turn carries the contents of the floppy disk (if any is inserted in the floppy drive). An additional subdirectory, **server** may be available if the 4400 is equipped with an Ethernet PC card and the TCP/IP Option and if a server disk has been mounted. This disk is shown under **/io/server**.

- **/rapid**. This directory carries various subdirectories with example RAPID! programs, result files, coupling loss files and many more. You can use this area to locally save your files on the 4400.

NOTE

Access to a remote PC on the network requires NFS server software to be running on the PC; the server must grant at least read access for the directories and files. Please read section I/O Configuration for more details.

RAPID! basic file area

This area is made up by a selection field. You may select any file that has been opened or newly created before.
The RAPID! Basic File selection field is available in all RAPID! menus.
A simple push on the **Next File** softkey or a click on the selection field navigates the cursor to it.

RAPID Basic File /usr/wavetek/4400/rapid/test/bchtst.bas
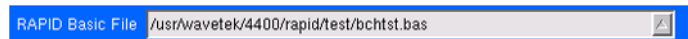
To select a file, simply click on the file name. Alternatively, push the **UP** and/or **DOWN** keys until the requested file is highlighted. Then confirm your selection by pushing the **ENTER** key.

Softkeys of the file menu

The file menu contains two sets of softkeys. The **ETC** softkey toggles between them.

| – ETC – | New File | Open | Save All | Close | Open Project | Next File |
| – ETC – | New Dir | Copy | Move | Erase | Print | |

New File — Creates a new RAPID! program file. You specify the name (and directory) in the File name – directory – selection area, which is activated automatically after you push this softkey.
As soon as the file has been created, it is added to the RAPID! Basic File list by the 4400 and you may start editing any time.

NOTE

As this softkey gives access to those three entry fields, you may also use this softkey if you want to e.g. set a new filter for the files to be displayed within the browser area (see section File name – directory – selection area for details).

Open — A push on this softkey opens the file currently highlighted in the Browser area. It also adds it to the RAPID! Basic File list.
After a file has been opened, you may edit, run or debug the file.

Save All — Saves all currently open files.
If there have been no changes made to a file since it was saved last, the file will remain unchanged.

Close — Closes the file currently selected on the RAPID! Basic File selection field list and removes it from the list.
If there have been any changes made to that file since it was saved last, you will be prompted to save the file.

Open Project — This softkey works similar to the **Open** softkey. Unlike the **Open** softkey, this softkey does not only open the file selected (highlighted) but also all basic files chained to this one.
All files will be added to the RAPID! Basic File list.

Next file — Pushing this softkey will activate the RAPID! basic file selection field and allow you to select one of the currently open RAPID! basic files.

New Dir — Creates a new subdirectory in the directory currently shown in the File Name – Directory – Selection display field.
The 4400 will prompt you with an entry field to enter the name of the subdirectory to be created. To enter the name, use the external keyboard or the typing tabs.

Copy — A push on this softkey allows you to copy the currently selected (= highlighted) file. Additional softkey labels will be displayed that allow the following choices:

> **OK –** Copies the currently selected file. The 4400 will prompt you with a dialogue box that allows you to enter the destination file name (including directory). To enter the file and directory name, use the external keyboard or the typing tabs.

> **to IO –** Copies the currently selected file to the floppy inserted in the 4400's floppy drive. The copy of the file will have the same name as the original file.

> **to RAPID –** Copies the currently selected file to the 4400's internal hard disk drive. The test set will prompt you with a dialogue box that allows you to enter the destination file name (including directory). To enter the file and directory name, use the external keyboard or the typing tabs.

> **Cancel –** Quits the copy function and takes you back to the second set of softkeys.

Move — With the help of this function, you may rename the file currently selected in the browser area.
The 4400 will prompt you with a dialogue box that allows you to enter the destination file name (including directory). To enter the file and directory name, use the external keyboard or the typing tabs.

Erase — Deletes the file currently selected on the RAPID Basic File selection field. Before the file is deleted, you will be prompted to confirm or discard the erasure of the file.

Print — After pushing this softkey, the 4400 will prompt you with a print output message:
`Printing file ....`
`Is a Printer connected to LPT, switched on and online?`
If you confirm by pushing **YES**, the 4400 will print the file highlighted (without any formatting).
Softkey **NO** aborts the printing process.

> **NOTE**
>
> The settings for the printer are performed in the `I/O` menu of the configuration section. To gain access, return to the Welcome menu, push the **TOOLS** key and then the [I/O] marker tab. Settings for the parallel printer port are performed in the section marked with `Parallel Port`.

Edit menu For program developers, this is probably the most important menu of the RAPID! environment.
The Edit menu is the menu you actually write programs in, before they can be run or debugged.
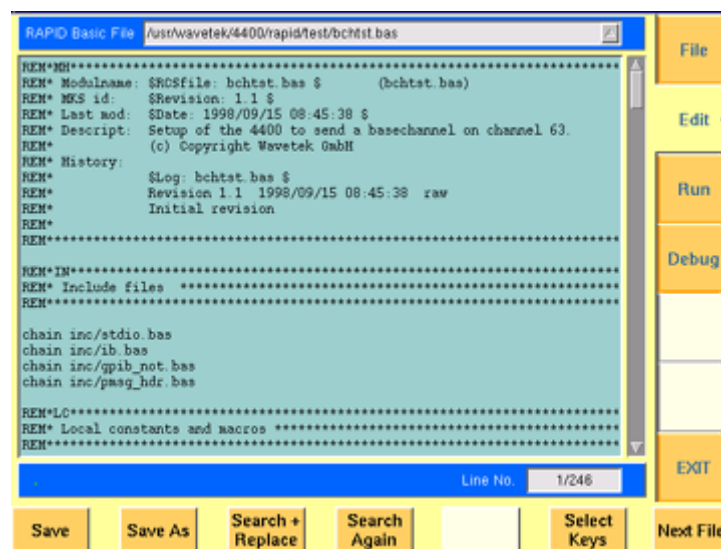
The Edit menu is subdivided into three main sections:

– RAPID! basic file area. Here you select the file you want to edit.
– Edit area, where the editing is carried out.

– Status area. This is where the 4400 displays error messages or gives the current cursor position in relation to the entire file.

The softkeys provide access to important functions of the Edit menu, like the search for text.

The marker tabs give instant access to the various menus of the RAPID! environment.

Edit area   The Edit area is where you create and edit program code, using the built-in BASIC-type language of RAPID!.

```
REM*MH*********************************************************************
REM* Modulname: $RCSfile: bchtst.bas $          (bchtst.bas)
REM* MKS id:    $Revision: 1.1 $
REM* Last mod:  $Date: 1998/09/15 08:45:38 $
REM* Descript:  Setup of the 4400 to send a basechannel on channel 63.
REM*            (c) Copyright Wavetek GmbH
REM* History:
REM*            $Log: bchtst.bas $
REM*            Revision 1.1  1998/09/15 08:45:38  raw
REM*            Initial revision
REM*
REM**********************************************************************

REM*IN*********************************************************************
REM* Include files  ******************************************************
REM**********************************************************************

chain inc/stdio.bas
chain inc/ib.bas
chain inc/gpib_not.bas
chain inc/pmsg_hdr.bas

REM*LC*********************************************************************
REM* Local constants and macros ******************************************
REM**********************************************************************
```

To work on program code, we strongly recommend the use of an external keyboard.
In principle, it is possible to do without, using the typing tabs, but this is a rather time-consuming way.

An external keyboard not only provides mapping of the 4400 keys. It also allows you to use the standard 'control codes' as with any usual text editor.
Some of the standard control codes are listed in the table below.

Ctrl+a	Select all
Ctrl+c	Copy
Ctrl+x	Cut
Ctrl+v	Paste

Status area

Line No.   1/246

In the status area, the 4400 shows the following information:

– Syntax errors – these are mainly typing errors with commands or missing command elements and prevent a RAPID! program from being executable.

– Runtime errors. These errors occur during runtime of a RAPID! program and usually lead to an unexpected end of the program run.

– The display field `Line No.` gives both the number of the line, the cursor is currently located in and the total number of lines of the file.

Softkeys of the Edit menu The Edit menu provides a number of file-handling functions that can easily be accessed through the softkeys.

| Save | Save As | Search +
Replace | Search
Again | Select
Keys | Next File |
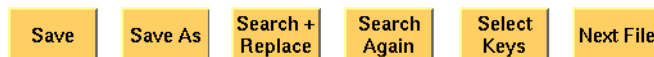
Save — A push on this softkey saves the currently open file without changing its name.

Save As — Saves the currently open file under a new name. After a push on this softkey, the 4400 will display a dialogue box that allows you to enter the destination file name (including directory). To enter the file and directory name, use the external keyboard or the typing tabs.
Also, the labels of the softkeys change and the 4400 will offer three new softkeys:

– **Save** When you push this softkey, the 4400 will save the current file under the file name entered in the dialogue box.

– **Protect** When you push this softkey, the 4400 will save the current file under the file name entered in the dialogue box in protected mode. This means that the file will be completely inaccessible to any editing attempts.

⚠️ **WARNING**

There is no way to edit a protected file again. Before selecting this function, please be sure that there is an unprotected copy of this file available.

– **Cancel** Quits the function and takes you back to the Edit menu with its basic softkeys.

Search+Replace — Enables you to find text passages and optionally replace them with other ones.
The text the 4400 shall search for is entered in the Search/Replace menu.

Search Again — Continues the search of the text passage as entered in the Search/Replace menu before. The direction of the search is "downwards", i.e. from the current cursor position towards the end of the file.

Select Keys — Changes the marker tabs to typing tabs. While the typing tabs are active, the label of this softkey changes to **Deselect Keys**. A push on that softkey brings back the basic marker tabs of the RAPID! environment and the initial softkey label.

Next File — Activates the RAPID Basic File selection field and allows you to select a different file. After you confirm your choice, the newly selected file is then displayed in the Edit area.

Search/Replace menu In this menu, you enter the text to find. Optionally, it can be replaced by new text, entered in the same menu.
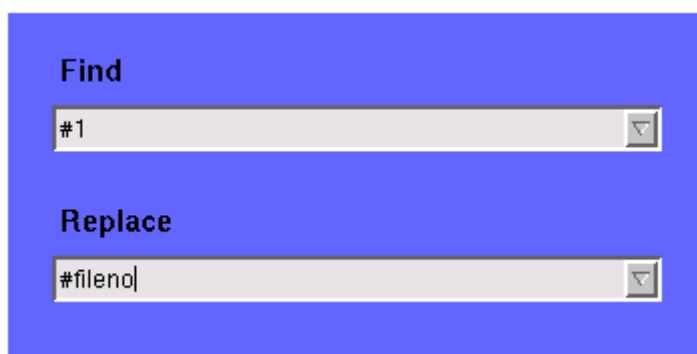
The text to be found is entered in the upper line (`Find`).
If the text found shall be replaced by another text, enter the new text in the lower line of this menu (`Replace`).
To move the cursor from the upper to the lower selection field and vice versa, use the **PGUP** and **PGDN** keys.
To get back to any text entered previously on these selection fields, push the **Select Prev.** softkey and the usual drop-down menu of a selection field will appear.

Softkeys of the Search/Replace menu

Find — Searches for the text entered on the `Find` selection field. The direction of the search is "downwards", i.e. from the current cursor position towards the end of the file.

Replace — Searches for the text entered on the `Find` selection field and replaces it with the text entered on the `Replace` selection field. The direction of the search is "downwards", i.e. from the current cursor position towards the end of the file.

Replace All — Searches for all occurrences of the text entered on the `Find` selection field and replaces every single one with the text entered on the `Replace` selection field. The direction of the search is "downwards", i.e. from the current cursor position towards the end of the file.

Select Prev. — Calls the usual drop-down menu of the currently active selection field and allows you to select any text entered previously.

Cancel — Quits the Search/Replace menu and takes you back to editing.

Typing tabs   In case an external keyboard is not accessible, there is still the built-in 'typewriter' of the 4400 available. The marker tabs change to typing tabs and allow to enter and/or edit text in a way similar to an SMS on a mobile phone.
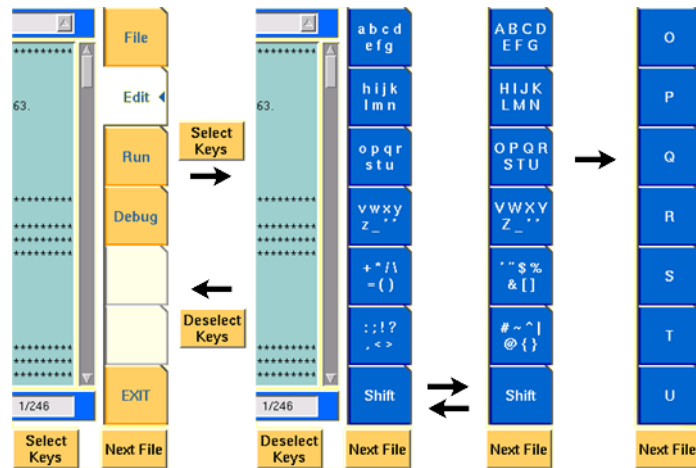
As soon as you push the **Select Keys** softkey, the marker tabs change to groups of letters and symbols.

The lowest marker tab changes to [Shift] and allows you to toggle between lower and uppercase letters.

As soon as you select a group of letters or symbols by pushing the corresponding marker tab, the marker tabs will change again to single letters or symbols. Push the marker tab of the desired letter or symbol. That one will appear on the entry field or edit menu and the marker tabs will change back to groups of letters and symbols automatically.

NOTE

Entry fields can also be edited using the **BACK** function key.

NOTE

Softkey **Deselect Keys** takes you back to the usual marker tabs of the RAPID! environment.

NOTE

Use the numeric keypad to enter numerals, the minus sign (-) or a period (.).

Run menu   The Run menu is where programs are started. If there are no syntax or runtime errors encountered, the program will smoothly run until it reaches its end and will display its status messages and comments in this menu.
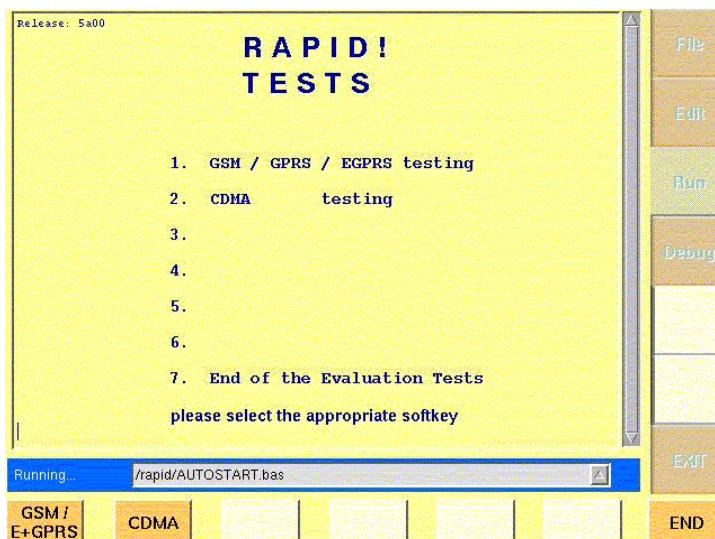
The Run menu is subdivided into two main sections:

– RAPID! basic file area. Here you select the file you want to run.

– Output area where the output of the running RAPID! program is displayed.

The softkeys provide access to important functions of the Run menu, like starting a RAPID! program.
The marker tabs give instant access to the various menus of the RAPID! environment.

NOTE

As long as no file has been opened or newly created, the marker tabs are grayed out and cannot be selected.

Output area   In the output area, the 4400 displays the output and status messages created by a running RAPID! program.
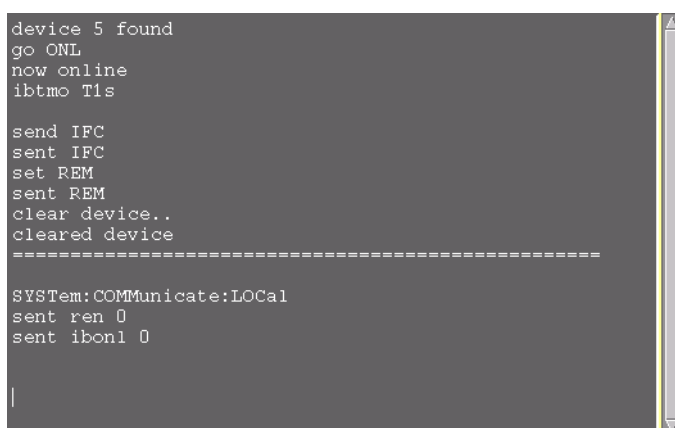
Softkeys of the Run menu   The Run menu offers two softkeys:

RUN — The RAPID! file, currently selected on the RAPID Basic File selection field, will be loaded and the program execution started.

Start File — Takes the cursor to the RAPID Basic File selection field and thus allows you to select the required file from the list. Next time you push the |RUN| softkey, this newly selected file will be executed.

Debug menu   The Debug menu provides you with a number of powerful tools in order to support your program development. The Debug menu consists of four areas that provide you with full information on different aspects of the program run.
In order to work with this menu in an efficient way, we strongly recommend the use of an external keyboard and a mouse.
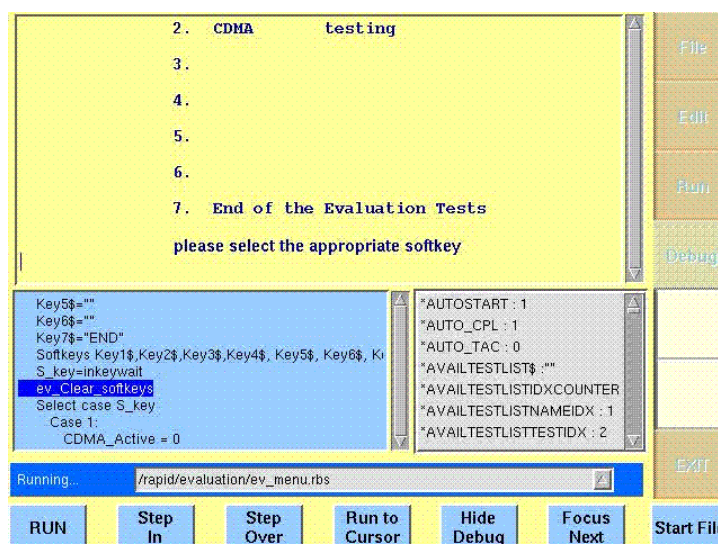
In this Debug menu only, the softkeys of the 4400 appear in blue. Thus, they can be easily distinguished from program-generated softkeys, as those appear in standard orange.

The areas are:

– RAPID! basic file area. Here you select the file you want to run.

– Output area where the output of the running RAPID! program is displayed.

– Source area. Here, the 4400 shows the program code being currently executed.

– The Variable area displays all program variables as well as their current value.

The softkeys provide access to important functions of the Debug menu, like starting a RAPID! program.
The softkeys of the Debug menu appear on blue background. Thus, they can be

easily distinguished from the program-generated softkeys.
The marker tabs give instant access to the various menus of the RAPID! environment.

NOTE

As long as no file has been opened or newly created, the marker tabs are grayed out and cannot be selected.

Source area

In the Source area, the 4400 displays the program code being currently executed. The program line highlighted in blue is the line that either has just been executed (in RUN mode) or the line that will be executed next (in STEP mode).

```
param(1)      = Modulationsart
param(2)      = BT
param(3)      = NormalHub

STOP
OUT_sendMsg SIG_DUPO, MOD_TARGET, MOD_PARAN

if DEBUG = TRUE then
  print "   Anzahl parameter:";AnzParam
```

To browse through the program, simply use the external keyboard or the cursor keys and function keys.

It is not possible to edit the program in this area. In order to do so, please use the Edit menu.

To run a program or to execute it step by step, please use the appropriate softkeys of the Debug menu.

In the Debug menu, only the first screen line of continued program lines will be highlighted.

Variable area

In the variable area, the 4400 displays all variables used in the RAPID! program run, including their current values.
While a program is e.g. waiting for an input, you may access the Variable area with the mouse and have a look at all current variables.

```
BT : 0.3
CALL_NETW_STAT : 132
CAL_HF_REC_RANG : 97
CAL_HF_REC_RANG_CLE : 98
CMD$ :"Start : 11:08:42"
DEBUG : 0
DEBUG_MESSAGE : 21
```

Altering the value of variables directly

Within the variable area, the values of the variables may be altered directly:

1   Select the variable with the mouse. The 4400 indicates the selection of the variable by highlighting it in dark blue.

> **Edit Watch: CMD$**
>
> Start : 11:08:42|

2   Enter the new value for the selected variable in the `Edit Watch:` dialogue box, using the 4400's numeric keys or the external keyboard.

3   Confirm the value entered with a push on the **ENTER** function key on the 4400 or a push on the external keyboard's **Enter** key.

4   The new value will now be used when you run or step the program using the corresponding softkeys of the Debug menu.

Alternatively

1   Open the Variable area by pushing the **Focus Next** softkey until the Variable area gets highlighted.

2   Select the variable you would like to alter with the cursor (simply use the cursor keys to do so). The currently selected variable is indicated by a thin black frame.

3   Activate the variable editing mode by confirming your selection with a push on the **Enter** function key on the 4400 or a push on the external keyboard's **Enter** key.
The 4400 indicates the selection of the variable by highlighting it in dark blue.

> **Edit Watch: CMD$**
>
> Start : 11:08:42|

4   Enter the new value for the selected variable in the `Edit Watch:` dialogue box, using the 4400's numeric keys or the external keyboard.

5   Confirm the value entered by pushing the **Enter** function key on the 4400 or by pushing the external keyboard's **Enter** key.

6   The new value will now be used when you run or step the program using the corresponding softkeys of the Debug menu.

Note: The variable selected last will remain highlighted until you select a new variable or run a new program.

Softkeys of the Debug menu

The Debug menu provides a number of specific debugging functions that can easily be called with the help of the softkeys.
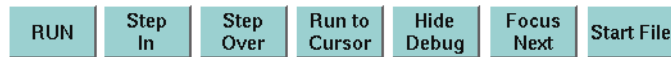
RUN — Loads the program currently selected on the RAPID Basic File selection field and starts execution of the program.
In this mode, the 4400 will highlight the **last** command performed in the Source area.

Step In — Executes the program line highlighted.
In case this line contains a GOSUB, CHAIN, SUB or FUNCTION call, the debugger steps in, i.e. it will follow the program branching.
In case the line contains an INKEY command, the program will be halted and wait for the push of a softkey. After the softkey was pushed, the program-defined softkey label will be displayed.
In this mode, the 4400 will highlight the **next** command to be performed in the Source area.

Step Over — Executes the program line highlighted.
In case this line contains a GOSUB, CHAIN, SUB or FUNCTION call, the debugger steps over it to the following line, i.e. the program branching will be ignored.
In case the line contains an INKEY command, the program will **not** be halted and will assume that no softkey was pushed.
In this mode, the 4400 will highlight the **next** command to be performed in the Source area.

Run to Cursor — Executes the program from the line currently highlighted to the line where the cursor is located.

Hide Debug — Hides the source and variable area to show the complete output area.
The label of this softkey will change to **Show Debug**. A push on **Show Debug** will display the hidden areas again.
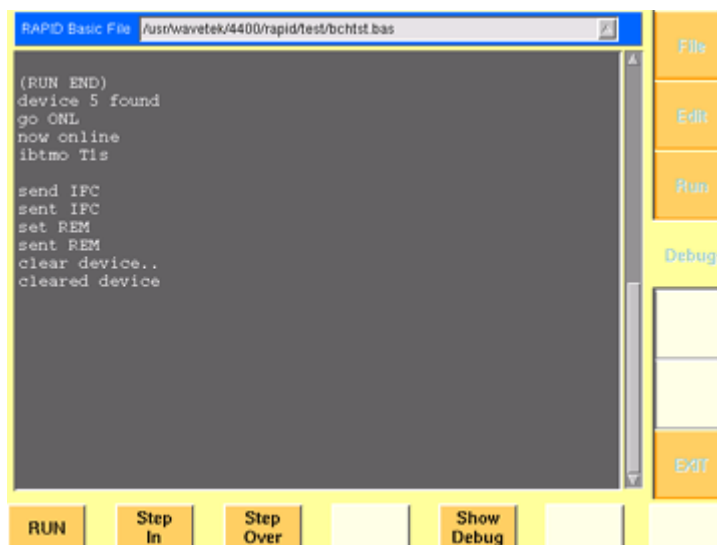
Focus Next — Toggles the cursor location between the output, the source and the variable area .

Start File — Activates the RAPID Basic File selection field and allows you to select a different file. After you confirm your choice, the newly selected file is ready for debugging.

RAPID! syntax

This chapter gives an overview of the RAPID! syntax and is subdivided into the following sections:

- General syntax – Here you will find all about the general aspects of the syntax like program line formats, basic rules, syntax check or notation.
- Program lines – This section outlines all details on the program line format and its elements.
- Variables – Are you familiar with numeric variables, string and array variables or dimensions? Here you will find it all.
- Constants – This section gives all information on numeric and string constants.
- Operators – RAPID! provides you with a number of operators like sign, arithmetic operators, comparisons or boolean operators.
- Expressions – These are combinations of variables, constants, function results and/or operators.

General syntax

BASIC form   The RAPID! programming syntax is compatible with current BASIC derivates, such as e.g. Quick BASIC or Visual BASIC.
However, RAPID! in its current form does not allow for object-oriented extensions.
On the other hand, RAPID! supports a number of functions that are vital for the use of the programming language on the 4400.

Program and line formats   Line labels (i.e. numbers) are optional. This means, they are not necessary.

A program line may include several BASIC commands. In that case, they have to be separated by colons (:).

The maximum length of a program line is 254 characters.
However it is suggested to keep the program lines short. This increases serviceability and eases debugging of a program.

The length of a program is (in theory) only limited by the disk space available.

Basic rules   The RAPID! program interpreter is not case-sensitive. This means that commands as well as constants and variables may be written in both uppercase and lower-case notation.

RAPID! commands and reserved words like `GET` or `GOTO` are not permitted as line labels or as names for constants and variables.

RAPID! commands, names of constants and variables as well as numeric values **must not** include blanks (spaces).

Syntax check   The syntax check is performed by the built-in interpreter.
In other words, after the start of a program, every single line is interpreted one by one and the corresponding action is initiated.
If the built-in interpreter comes across a command in a program line that it can't understand, then this is regarded a syntax error.

> **NOTE**
>
> Most of the syntax errors are simple misspellings of commands or typing errors.

The interpreter always checks a program after it was started. A detected syntax error causes the program to be aborted and leads to an error message indicating the erroneous program line.

Notation   To explain the language syntax, the following notation is used:

`[item]`

   Square brackets indicate an optional item, which can also be omitted.
   Example: `[sign] number` is a number with an optional + or – sign.

`item | item`

   Vertical bars separate entries of a list and indicate that precisely one element must be used.
   Example: The following list contains the four seasons `spring | summer | fall | winter`. One of these entries gives the current season.

`{item}`

   Brackets enclose an item or a list of items, which can be repeated several times.
   Example: `{ A...Z }` is the notation for text.

`item1 := item2`

   The colon followed by the equality symbol gives the definition of an item.
   Example: `binary digit := 0 | 1` defines the digits of a dual number system in a way that a 'binary digit' can either be '0' or '1'.

> **NOTE**
>
> Notations can also be nested. This makes them suitable for expressing even complicated syntax.

Example: `boolExp := NOT boolExp | boolExp AND | OR | XOR boolExp`
This means by words: A boolean expression is either the negation of another boolean expression, or two boolean expressions linked with one of the three operators: AND, OR, and XOR.
Please note that this is only an example. The complete definition of Boolean expressions is a bit more complex.

Program lines

Each RAPID! program consists of a number of program lines. Those lines contain the BASIC statements and have the following structure:

`[label:] statement [: statement]`

`label:`

A program line starts with an optional label. In contrary to AUTORUN programs, line numbers are not required.
With RAPID!, labels are mainly used as branch targets.

A label must always end with a colon.

`statement`

A statement is a BASIC command.

Important notes

– A program line always needs to be terminated by a carriage return (CR). This symbol is equivalent to pushing the 'Enter' key at the end of a program line.

– It is not possible to show the maximum of 254 characters in one line on-screen. Therefore, a program line may stretch out over several lines on-screen. To indicate that the program line is continued, put a blank followed by an underscore ' _' and then push **Enter** to start a new line.
Example: `print "RAPID!" _`
`;" Program"`

– There must not be empty lines between continued program lines.

– An underscore at the end of a 'REM' command will cause a syntax error.

– The debugger will only highlight the first screen line of continued program lines.

– As mentioned before, a program line may contain up to 254 characters and several statements. However, we strongly recommend to use only one statement per program line wherever possible. Several statements in a program line lower the readability of the program code and usually make servicing a program quite tricky.

Variables   Variables are used to save data for later use. There are two types of variables: numeric variables and string variables.

– **Numeric variables** contain numbers only (0...9) and are saved in in the so-called double (floating point) format on the 4400.
This is an eight-byte value with a range from –1.79769313486232E308 to –4.94065645841247E–324 for negative values and from 4.94065645841247E–324 to 1.79769313486232E308 for positive values.

– **String variables** may contain any combination of numbers (0...9) and letters (A...Z). Their name always must end with the '$' symbol.
The length of a string variable (i.e. the number of characters) may range from 0 to 65,535 characters.

Variables in the RAPID! environment are always local. This means that a variable is only valid within its own RAPID! program.
When you chain programs, a local variable is not available to the program chained. If a variable shall be global (i.e. it is available to all chained programs), simply specify it as global.

The main commands dealing with variables are DIM, GLOBAL, VARIABLE and ERASE.

Arrays

– All RAPID! variables can make up arrays of up to two dimensions. The index value per dimension may range from –32,765 to +32,765.

> **NOTE**
>
> An array is similar to a table or a spreadsheet.

Syntax

A variable name is not limited in length, but only the first 20 characters are significant. A variable name has the following syntax:

– `A...Z | _{0...9 | A...Z | _} [( index1 [, index2] )]` for numeric variables and

– `A...Z | _{ 0...9 | A...Z | _}$ [( index1 [, index2] )]` for string variables.

Notes

– The first character of a variable name has to be a letter.

– Uppercase and lowercase letters have the same meaning with RAPID!.

– A...Z are uppercase or lowercase letters.

– 0...9 are numbers between 0 and 9.

– `index1` and `index2` are integer values, making up an array.
The range of an integer value is from –32,765 to +32,765.

Examples

– `1_stValue` is not a valid variable name as it starts with a number.

– `myvar_1`, `MyVar_1` or `MYVAR_1` will be understood by RAPID! as being the same variable.

- `StringArray_of_Dimension_2$(1,1)` defines a valid string variable array. However, due to the length of its name, RAPID! will not distinguish it from `StringArray_of_Dimension_1$(1,1)`.

Physical units

Numeric variables and expressions can have physical units. Depending on the group they belong to, the units of those variables and expressions may be:

Group Name	Physical units
Frequency	Hz, kHz, MHz, GHz
Power	dB, dBm, µW, mW, W
Angles	DEG or °, RAD
Percent	%

Notes

- Percent values are treated as unit-less when arithmetical operations are performed.
- Additions, subtractions and comparisons may only be performed within a group as specified above.
- Divisions through a value containing a physical unit of the same group will lead to a result without physical unit.
- When multiplying values, only one factor may have a physical unit.

Constants

Constants are used to store numeric or string values, which need to be kept unchanged during runtime of a RAPID! program.
Typical examples are common mathematical constants like 'pi', the power loss factor of a cable used for testing or the type code of a mobile phone.

Numeric constants

A numeric constant is a decimal, binary, octal or hexadecimal value with an optional physical unit:

```
decimal | binary | octal | hexadecimal [physical
unit]
```

Type of constant	Syntax	Example	Limits
Decimal	{0...9} [.{0..9}]	123.456	$-1.79769313486232E308$ to $-4.94065645841247E-324$ for negative values and from $4.94065645841247E-324$ to $1.79769313486232E308$ for positive values
Binary	&B {0...1}	&B100111011	2 to the power of 32 (2^{32})

| Octal | &O {0...7} | &O37721 | 2 to the power of 32 (2^32) |
| Hexadecimal | &H {0...9 \| A...F} | &H4AF3 | 2 to the power of 32 (2^32) |

Names of constants without any identifier (&B, &O, &H) are always interpreted as double (floating point) values.
These are eight-byte values with a range from −1.79769313486232E308 to −4.94065645841247E−324 for negative values and from 4.94065645841247E−324 to 1.79769313486232E308 for positive values.

String constant

The syntax of a string constant is as follows:

```
" {any character} "
```

A string constant may contain any character accessible via the (external) keyboard and is always framed by quotation marks.

> **NOTE**
>
> If you want a quotation mark to appear within a string, simply enter two consecutive quotation marks.
>
> **Example:** `"Willtek 4400 ""RAPID!"" Program"` equals the following contents of the string variable: 'Willtek 4400 "RAPID!" Program'

> **NOTE**
>
> The maximum length (i.e. number of characters) for a string constant is 65,535.

Operators The following table lists the numeric, string and boolean operators, along with their priority. The smaller the number in the priority column, the higher the priority.
During calculations, first all operators with priority 1 are performed, than all of priority 2 and so on. Priorities assure that the usual sequence of operations (e.g. multiplication and division first, then addition or subtraction) are performed without the need of brackets.

Type	Operator	Priority	Meaning
Sign operators	+	1	Positive sign
	−	1	Negative sign
	NOT	1	Negation (Inversion)
Arithmetic operators	*	2	Multiplication
	/	2	Division
	\	2	Integer division
	MOD	2	Modulo division

	+	3	Addition
	-	3	Subtraction
Comparison opera-tors	=	4	Equality
	<>	4	nonequality
	<	4	Less
	>	4	Greater
	<=	4	Less or equal
	>=	4	Equal or greater
Boolean operators	AND	5	Logical AND
	OR	5	Logical OR
	XOR	5	Logical XOR

NOTE

In case of an integer division or a modulo division, the two operands are first converted into integer values, before the operation is performed. This means, that the operands will be rounded before any operation takes place. The rounding procedure will follow usual 'commercial' rules.
Example: The result of (5 MOD 2.3) will be 1 as it is interpreted as (5 MOD 2) by RAPID!.

NOTE

Regardless of the format of the operands with comparison operators, the result will always be a boolean value, i.e. number 0│1.

How to use operators

– Numeric expressions may use any operator.

– Boolean expressions, which are in fact numbers (0 or 1), may use any operator.

– String expressions may use all comparison operators and the arithmetic addition.
Example: "Willtek " + "4400" will result in the string expression "Willtek 4400".

Expressions   An expression is a valid combination of variables, constants, function calls and operators.

Basically, there are three types of expressions:

– numeric expressions

– string expressions

– boolean expressions

Numeric expressions

```
numExp := [signOp] numVar | numConst | numFct |
boolExp | numExp [ operator numExp ]
```

Where

- `signOp` is the sign operator (+ or –),
- `numVar` is a numeric variable,
- `numConst` is a numeric constant,
- `numFct` is a BASIC function that returns a numeric result,
- `boolExp` is a boolean expression, as explained below

━━━▶ | **NOTE**

A numeric expression always delivers a numeric result.

Example: `OneVar + ( LEN (AString$) – 2) * 5`
This is a numeric expression that contains

- a numeric variable (`OneVar`),
- a function (`LEN`) of a string variable (`AString$`),
- the operators +, – and *
- and two numeric constants (`2` and `5`).

The result of this expression will be a numeric value.

String expressions

```
stringExp := stringVar | stringConst | stringFct { +
stringVar | stringConst | stringFct }
```

Where

- `stringVar` is a string variable,
- `stringConst` is a string constant,
- `stringFct` is a BASIC function that returns a string as the result.

Note: A string expression always delivers a string.

Example: `FirstName$ + " " + LastName$` is a string expression. It delivers back a string that contains the two string variables `FirstName$` and `LastName$`, separated by a blank.

Boolean expressions

Boolean expressions are more complex than the preceding ones:

```
boolExp := boolOperation | comparison | boolFct
```

Where

- `boolOperation := NOT boolExp | boolExp AND | OR | XOR boolExp`
- `comparison := stringExp compOperator stringExp | numExp compOperator numExp`

– `compOperator := < | <= | = | >= | >`

– `boolFct` is the return value of a boolean function.

> **NOTE**
>
> A boolean expression always delivers either number 0 (for false) or 1 (for true) as the result. You may perform arithmetic operations with these results.

Example: `(Country$="UK") AND (NetworkPrefix>=262)`
In case that string variable `Country$` equals "UK" and at the same time the numeric variable `NetworkPrefix` is equal to or greater than 262, then this boolean expression will deliver 1 as the result. In any other case, the result delivered will be 0.

Commands

The language elements of RAPID! are BASIC commands and BASIC functions. In order to access the internal registers of the 4400, the SCPI commands are available which are described in detail in section SCPI.

– Commands make up a RAPID! program. They are used to tell the 4400 what it is to do.

– Functions perform specific predefined tasks and provide values, that can be used within any RAPID! program.

– In BASIC, you may place a single command in a program line. However, placing just a single function in a program line will lead to an error.

Overview   Dependent on their general functions, the BASIC commands of RAPID! are subdivided into six groups:

– General commands are used to control the overall aspects of a program. Typical general commands are e.g. CHAIN, REM, END, WAIT or STOP.

– Screen commands control the screen output of the 4400. They comprise CLS, TEXTATTR, LOCATE, INPUT, PRINT/OUTPUT and SOFTKEYS.

– Commands related to variables, procedures and functions. These are DIM, GLOBAL, VARIABLE, ERASE, SUB, FUNCTION.

– Control commands determine the flow of the program. There are two sorts of control commands: loops (FOR-NEXT, DO LOOP, WHILE, UNTIL) and conditions (IF, CASE).

– Branch commands, such as GOTO, GOSUB and ON ERROR allow to change the continuous sequential flux of a program and to jump to specific areas of a program.

– Commands for input/output handling allow special input/output tasks. They are OPEN, CLOSE, INPUT, PRINT and OUTPUT.

General commands　　General commands are used to control the overall aspects of a program. The following general commands are available in RAPID!:

Command	Short description
CHAIN	Connects additional RAPID! programs to the main program.
END	Ends the program.
LET	Assigns an expression to a variable.
REM or '	Comment.
STOP	Debugger breakpoint.
WAIT	Halts the program for a specified amount of time.

CHAIN

Syntax　　CHAIN stringExp

Parameters　　stringExp is a string expression. It must be identical with an existing RAPID! program file name.

Description　　Loads the specified program and executes any command lines which come before the first subroutine or function. When this is complete, control is returned to the statement following the chain command.
Global variables of the initially calling program can be used by all chained programs.
Subs, functions and labels are global by definition and can be used by all programs (i.e. the initially calling program and all chained programs). Consequently, all names of subs and labels have to be unambiguous.
Note: Subs and functions of the chained program will be ignored and **not** run if not explicitly called (see example below).
CHAIN is a perfect tool to split large programs into logical units or to build a library with modules for specific tasks and functions. If you do so, please keep the following hints in mind:
– In the chained program, first declare the variables required as global,
– then write the required program code as a function or sub.
– Use the chain statement at the beginning of the main program. This will result in the 4400 running the program code of the chained programs (with the exception of the functions themselves). Thus, the 4400 will initialize all required global variables for later use.
A chained program may also chain other programs down to a depth of four nested CHAINs.

Example   Contents of main program:
```
PRINT "Now calling chained program"
CHAIN "TESTPROG.BAS"
PRINT "Back again to main program"
testsub
```

Contents of TESTPROG.BAS:
```
PRINT "Hello from chained program
TESTPROG"
SUB testsub()
PRINT "Subroutine called"
END SUB
```

Execution of the main program will result in the following screen output:
```
Now calling chained program
Hello from chained program TESTPROG
Back again to main program
Subroutine called
```

END

Syntax   END

Parameters   END does not have any parameters.

Description   Ends the program.

Example
```
IF pwr > 2.2 then END
```
In case the value of variable `pwr` exceeds 2.2, the program will be ended immediately.

LET

Syntax
```
[ LET ] numVAR = numExp | stringVar =
stringExp
```

Parameters   `numVAR` is a numeric variable,
                   `numExp` is a numeric expression,
                   `stringVar` is a string variable,
                   `stringExp` is a string expression.

Description   Assigns the value of a (numeric or string) expression to a (numeric or string) variable.
                   Note: `LET` is optional and can be omitted.

Example
```
TheString$ = "This is a ""String"": +
Inside$ + ", in case anybody doubts
it."
```

REM or ' (Comments)

Syntax   REM anyText or ' anyText

Parameters		`anyText` can be any text not exceeding the current line.
		Note: The program line extension symbol can't be used on REM lines.
Description		Enables you to include comments in the programs. Comments not only help other people to maintain your code; it might well be that you need some hints to understand your own code when you look at it from a long distance in time.
Example		`REM The next line will define a variable` `LET i = 42 ' Who does not call his integers "i"`

STOP

Syntax	`STOP`
Parameters	`STOP` does not have any parameters.
Description	Stops the program execution, but does not end the program (important e.g. to keep the values of variables when debugging a program).
Example	`If n = 10 then STOP`

WAIT

Syntax	`WAIT msec`
Parameters	`msec` specifies the time the program will wait in milliseconds. `msec` can only be entered as an integer value. The maximum time is 32,767 milliseconds.
Description	Stops the program execution for the duration of the time specified in parameter `msec`.

Example	`PRINT "Please connect mobile now"`
	`WAIT 2000`
	Prints user information and then halts the program for two seconds.

Screen commands

These commands control the screen output of the 4400.
The following screen commands are available in RAPID!:

Command	Short description
CLS	Clears the screen.
INPUT	Waits for a user entry and stores the value.

LOCATE	Sets the cursor position on-screen for the subsequent output command.
PRINT or OUTPUT	Shows an expression on-screen.
SOFTKEYS	Labels the 4400's softkeys.
TEXTATTR	Sets background color and text size, font and color.

CLS

Syntax   CLS [textCol] [,[bgCol] [,fontSize[b] [i]]]

Parameters   textCol is the new text color,
bgCol is the new background color,
fontSize is the new font setting,
b stands for bold text,
i stands for italic text.

Description   Clears the screen and positions the cursor in the upper left screen corner.

Optionally, new background and text settings for the subsequent output operations may be specified:

Text and background color may be assigned one of the predefined constants BLACK, WHITE, GREY, MGREY, DGREY, RED, GREEN, BLUE, YELLOW, MAGENTA, CYAN, DGREEN, DCYAN, DBLUE, BROWN, PURPLE.

Alternatively, you may assign a code between 0..&HFFFFFF. This code represents an RGB model of the screen colors, where the first two hexadecimal figures represent the red color, the middle two figures the green and the last two figures the blue color.

The font setting is specified by the font code followed by the font size.

The following fonts are available (corresponding codes in parentheses):

Charter (char), Courier (cour), Courier 10 Pitch (courier), Dutch 801 (dutch), Helvetica (helv), Lucida (lu), Lucida Bright (lub), Lucida Terminal (lut) New Century Schoolbook (ncen) PC Sansserif (pcss), PC Serif (pcs), PC Terminal (pcterm), Swiss 721 (swiss), Symbol (symb) Technical (tech), Times (time) Useless (uselss), Utopia (utop) .

The selectable font sizes are 8, 10, 12, 14, 18, 20, 24, 32, 48 pt.

Optionally, the text can be set to bold (b) and/or italic (i).

Note: The font codes are string values, they have to be put in quotation marks (e.g. "cour14"). Otherwise, a syntax error will occur.

Examples	CLS (Clears the screen and keeps the current text and background settings.) CLS , , "cour14" (Clears the screen, keeps the colors and selects Courier 14 pt as new font.) CLS white, blue, "lu10bi" (Clears the screen and continues with white text on blue background, font Lucida 10 pt, bold and italic.)

INPUT

Syntax	INPUT [[inputPrompt$] ,variable]
Parameters	inputPrompt$ is a string expression, variable is a numerical or string variable.
Description	Causes the program to wait for user input, reads data entered on the keyboard into a variable and then continues the program run. INPUT will first output the inputPrompt$ on the 4400 screen. Then it waits for the user's input. INPUT can read data entered either into a numeric or a string variable. The 'signal' for input finished is the CR symbol (achieved by the user pushing the **ENTER** key). The symbols entered by the user will be stored in variable and the program will continue. **Notes:** – External keyboard and the keys of the 4400 may be used in parallel. This means that an entry on the external keyboard may be confirmed by pushing the 4400's **ENTER** key. – In case variable has been defined as a numerical variable and the user enters any other symbol than numbers and the decimal point, this will lead to an error. A simple trick to avoid that is to always read the value entered as a string and later convert that string into a numerical value: INPUT "Please enter channel number:" _ , channelno$ channelno = VAL(channelno$) – With the help of the INPUT command, you may also read data from a file or the SCPI system.
Examples	INPUT "What's your name? ", name$ INPUT (The program waits until the user pushes the Enter key.)

LOCATE

Syntax	LOCATE x, y
Parameters	x stands for the cursor's new horizontal position, y for the cursor's new vertical position.

Description
Sets the cursor to a new position on-screen. Any subsequent text output will start from that position.
Parameter `x` is counted in character positions rather than in pixels; parameter `y` in screen lines. The values for `x` and `y` are dependent on the currently set font size. However, the 4400 enables the user to scroll the screen output.
For practical reasons, `x` should be kept below 50 and `y` below 25.

Examples
```
IF (channelno > 2099) then LOCATE 24, 0
Print "Input out of range. " _
;"Please reenter."
```
(In case the channel number entered by the user is out of range, a prompt will appear in the 'status line' of this RAPID! program.)

PRINT (OUTPUT)

Syntax
```
PRINT [ { [USING usestr$,] exp [ , | ;
|TAB(number);] } ]or
OUTPUT [ { [USING usestr$,] exp [ , | ;
|TAB(number);] } ]
```

Parameters
`exp` is a numeric or string expression,
`usestr$` defines an output format for numeric data,
`number` stands for an integer figure, forcing the next `exp` of the same `PRINT` or `OUTPUT` command to appear on a specific position on-screen.

Description
Displays the specified expression(s) on-screen.
In case the separator is a comma (,), the expression following the current one will be displayed at the next tabulator position.
In case the separator is a semicolon (;), the expression following the current one will be displayed directly after the previous one.
`OUTPUT` is a 100% synonym for `PRINT`.
`PRINT USING "a.b"` allows to format numeric output. While `a` gives the total amount of digits to be used for the output (including decimal places, the decimal point and the sign), `b` specifies the number of decimal places (see example below).
Note: The `PRINT` command is also available to output data to a file or the SCPI system.

Examples	`PRINT var1, var2, var3` (Three numeric variables are displayed, separated by tabs). `OUTPUT "Here ";"you ";"go!"` (The phrase will be displayed as 'Here you go!'). `PRINT USING "10.2";21,5` (The value will be displayed using a total of 10 digits (including the decimal point and the sign). 2 digits will be used for the decimal places. The value will be printed as '`21.50`' with five leading blanks.) `OUTPUT TAB(10);9` (The number 9 will be printed on the tenth cursor position following the current one: ' 9'.)

SOFTKEYS

Syntax	`SOFTKEYS [str1$, [str2$, [str3$,` `[str4$, [str5$, [str6$, [str7$]]]]]]]`
Parameters	`str1$...str7$` are string expressions.
Description	Labels the 4400's softkeys. The labels specified are assigned to the softkeys from left to right. In case less than seven string expressions are used, one or more softkeys remain without label.

Examples	`SOFTKEYS "Key1","","Key3"`
	(The first and third softkey from the left are labelled and may be used for `INKEY` or `INKEYWAIT` commands.)
	`SOFTKEYS`
	(All softkeys remain unlabelled. There are no values that could be handed over to an `INKEY` or `INKEYWAIT` command, whatever softkey may be pushed. In case of an `INKEYWAIT` command, this will result in an endless loop.)

TEXTATTR

Syntax	`TEXTATTR [ textCol ] [, [bgCol ]` `[, fontSize [b] [i] ] ]`
Parameters	`textCol` is the new text color, `bgCol` is the new background color, `fontSize` is the new font setting, `b` stands for bold text, `i` stands for italic text.
Description	Selects new color and text settings for the subsequent output operations. For detailed parameter description, please refer to command `CLS` in this subsection.
Examples	`TEXTATTR red` (Text color set to red.) `TEXTATTR , cyan, "cour18"` (Courier size 18 on cyan background.)

Commands related to variables, procedures and functions

The following commands concern the declaration, output and deletion of variables and the usage of procedures and functions:

Command	Short description
DIM	Declares local variables.
ERASE	Releases allocated memory of a variable.
FUNCTION	Declares a function.
GLOBAL	Declares global variables.
SUB	Declares a procedure (subroutine).
VARIABLE	Prints variable contents to screen or file.

DIM

Syntax	`DIM varName [{, varName}]`
Parameters	`varName` is a valid variable name.
Description	Declares a local variable. The variable is only accessible to the program element it has been declared in. This means, a local variable declared in a function is not accessible by the main program and vice versa. Variables that are accessible for all program elements are called 'global' (please see below for more details). **Note:** There is no need to declare local variables. Within the RAPID! environment, all variables are interpreted as local by definition. However, after a variable has been declared using the `DIM` command, all subsequent variables have to be declared with `DIM` as well. In order to keep programs easily maintainable and to avoid errors due to misspelled variable names, we strongly recommend to declare variables in larger programs.
Example	`DIM Days$(7), Months$(12)` declares two string arrays.

ERASE

Syntax	`ERASE varName [ {, varName} ]`
Parameters	`varName` is a valid variable name.

Description	Deletes a variable and releases the memory used. This command is useful to keep memory consumption of a program as low as possible. Especially large variable arrays or long string variables should be erased as soon as they are no longer required. **Note:** Local variables are automatically erased at the termination of the related procedure or function.
Example	`ERASE Journal$`

FUNCTION – EXIT FUNCTION –
END FUNCTION

Syntax	`FUNCTION fnName [$] ( [ parameter`
	`[{, parameter } ]] )`
	`[ instructions ]`
	`fnName [$] = fnExp`
	`[ EXIT FUNCTION ]`
	`END FUNCTION`

Parameters	`fnName` is a valid name for the function (same syntax rules apply as for variable names), `parameter` may be any valid numeric or string expression, `instructions` are an (optional) number of program lines, `fnExp` is an expression (of the same data type as the function itself).
Description	Declares a function (`FUNCTION ... END FUNCTION`). Functions are commonly used to e.g. calculate the result of mathematical expressions. After a function has been declared, it may be called by its name `fnName`. A function call must contain all parameters in the same order as specified in the function's declaration. The return value of a function will either be a numeric expression (in case of `fnName`) or a string expression (in case of `fnName$`). A function may call itself recursively. All variables used in a function are local unless declared as `GLOBAL` (please see below for details). All parameters are treated as local. Using the `EXIT FUNCTION` command, you may leave any function at any place or condition (e.g. in case a value exceeds a certain limit, the function exits and passes on control to an error handling system).

Example The following example calculates the faculty (n! := 1*2*3*...*n) of a given integer number n. In order to do so, the function calls itself recursively:

```
FUNCTION Faculty(n)
IF n<2 THEN
Faculty = 1
EXIT FUNCTION
END IF
Faculty = n * Faculty(n-1)
END FUNCTION
```

GLOBAL

Syntax `GLOBAL varName [ {, varName} ]`

Parameters `varName` is a valid variable name.

Description Declares a global variable. Global variables are accessible in the main program, in all programs included by CHAIN and in all related procedures and functions.
Any global variable is valid and accessible from the program line, in which it has been declared.
Note: If a local variable has been declared or used before, that carries the same name as the global variable, then the global variable is not accessible.

Example `GLOBAL strg$, counter` declares a global string and a global numeric variable.

SUB – EXIT SUB – END SUB – CALL

Syntax
```
SUB subName ( [ parameter [{, parame-
ter } ]] )
[ instructions ]
[ EXIT SUB ]
END SUB
...
[CALL] subName ([ parameter [{,
parameter } ]] )
```

Parameters `subName` is a valid name for the procedure (subroutine) (same syntax rules apply as for variable names),
`parameter` may be any valid numeric or string expression,
`instructions` are an (optional) number of program lines.

Description	SUB ... END SUB declare a procedure.
	Procedures are commonly used to e.g. perform certain operations that have to be carried out several times during the program run.
	After a procedure has been declared, it may be called by [CALL] subName.
	A call of a procedure must contain all parameters in the same order as specified in the procedure's declaration.
	A procedure may call itself recursively.
	All variables used in a procedure are local unless declared as GLOBAL (please see above for details).
	All parameters are treated as local.
	Using the EXIT SUB command, you may leave any procedure at any place or condition (e.g. in case a value exceeds a certain limit, the procedure exits and passes on control to an error handling system).
Example	The following example defines a procedure that writes the current contents of all variables into a file with a given file name:

```
SUB VarsToFile(fileName$)
LET ff = FREEFILE
OPEN fileName$ FOR OUTPUT AS #ff
VARIABLE #ff
CLOSE #ff
END SUB
```

VARIABLE

Syntax	VARIABLE [#fileNo]
Parameters	fileNo is a handler of an open file.
Description	Outputs the contents of all currently used variables on-screen.
	The currently used variables are all variables that were defined or used before the VARIABLE command appeared in the program run.
	In case you specify the optional file handler (#fileNo), the output will be sent to a file.
	Note: During program development, it is sometimes quite useful to know the contents of all variables. The VARIABLE command also is a good tool to unmask misspellings with variable names.
Example	The following example program writes the contents of all variables to a file called 'VAR.TXT'.

```
OPEN "VAR.TXT" FOR OUTPUT AS #1
VARIABLE #1
CLOSE #1
```

Control commands

Control commands organize the process of a program. They e.g. allow to run a specific part of a program several times or to select between different sets of commands, depending on a condition.

In RAPID!, the following control commands have been implemented:

Command	Short description
DO ... LOOP	Loop with condition.
FOR ... NEXT	Loop with a specified number of cycles.
IF ... THEN	Checks a condition.
SELECT CASE	Selects one of several cases.

DO ... LOOP, WHILE, UNTIL

Syntax

Syntax 1
```
DO [ {WHILE | UNTIL} boolExp ]
[ instructions ]
[ EXIT DO ]
[ instructions ]
LOOP
```
Syntax 2
```
DO
[ instructions ]
[ EXIT DO ]
[ instructions ]
LOOP [ {WHILE | UNTIL} boolExp ]
```

Parameters

`boolExp` is a boolean expression, `instructions` are an (optional) number of program lines.

Description

Dependent on a condition (`boolExp`), this command structure repeatedly runs a program segment.

Syntax 1 checks the condition before the loop is run. If the condition is true in the moment the program arrives at the loop, it will not be run at all.

Syntax 2 checks the condition after the loop has been run.

`WHILE` means that the loop will be run as long as `bool-Exp` is true.

`UNTIL` will run the loop until `boolExp` becomes true.

`EXIT DO` causes the loop to be left instantly.

If the optional part with the condition is left out, the loop will be run endlessly. In this case, the only way out is an `EXIT DO` command somewhere within the loop to prevent the program from being trapped.

Examples   The three following examples perform the same task in different manners. They all print a text to the screen until a softkey is pushed:

```
DO
PRINT "Please push a softkey! ";
IF INKEY <> 0 THEN EXIT DO
LOOP
DO WHILE INKEY = 0
PRINT "Please push a softkey! ";
LOOP
DO
PRINT "Please push a softkey! ";
LOOP UNTIL INKEY <> 0
```

FOR ... NEXT

Syntax

```
FOR countVar = numStart TO numEnd [
STEP numStep ]
[ instructions ]
[ EXIT FOR ]
[ instructions ]
NEXT countVar
```

Parameters   `countVar` is a numeric variable,
`numStart, numEnd, numStep` are numeric expressions,
`instructions` are an (optional) number of program lines.

Description   Repeats a program segment a specified number of times. To do so, a counter (`countVar`) is used.
The initial value of the counter is stated in the `numStart` expression, while the final value is stated in the `numEnd` expression.
After each run, the counter is increased by `numStep`. If the optional `STEP` part is missing, the counter will be incremented by 1 per run.
After the loop was run, the condition (`countVar = numEnd`) will be checked. If the condition is false, the loop will be run again.
If the condition is true, the program will resume with the command following the `NEXT countVar` command.
In order to immediately exit a `FOR...NEXT` loop, use the `EXIT FOR` command.

Examples   The following example program prints the ASCII characters A...Z to the screen:

```
FOR i=65 to 90
PRINT CHR$(i)
NEXT i
```

The following example reverts the string Matter$ into the string AntiMatter$:

```
AntiMatter$ = ""
FOR i=len(Matter$) TO 1 STEP -1
AntiMatter$ = AntiMatter$ + MID$(Mat-
ter$, i, 1)
NEXT i
```

IF ... THEN, ELSE, ELSEIF, END IF

Syntax

```
Syntax 1
IF boolExp1 THEN instructions1 [ ELSE
instructions2 ]
Syntax 2
IF boolExp2 THEN
[ instructions3 ]
[ ELSEIF boolExp3 THEN
[ instructions4 ] ]
[ ELSE
[ instructions5 ] ]
END IF
```

Parameters

`boolEXP1/2/3` are boolean expressions, `instructions1/2/3/4/5` are an (optional) number of program lines.

Description

Depending on a boolean expression or condition (`bool-Exp`), the program will resume with different sets of commands.

There are two different forms of this command, indicated by 'Syntax 1' and 'Syntax 2' above.

Syntax 1 is the 'one-line' command. If `boolExp1` is true (or a numeric value is <> 0), the command(s) following `THEN` will be executed. If `boolExp1` is false (or a numeric value = 0), the commands following the optional `ELSE` will be executed.

Syntax 2 is the 'multiline' version of the command. This syntax should be used in case there are several commands to be executed, depending on the conditions.

In case `boolExp2` is true, `instructions3` will be executed.

If `boolExp2` is false, the subsequent `ELSEIF` command will be looked for. If this is true (i.e. `boolExp3` is true), `instructions4` will be executed.

There may be several `ELSEIF` conditions within one IF-clause.

Should all `boolExp2/3` be false, `instructions5` will be performed.

Examples

```
IF tired THEN PRINT "Go to bed!" ELSE
PRINT "Go shopping!"
```

```
IF (i <= 0) THEN
PRINT "Not a positive number"
ELSEIF (i MOD 2 = 0) THEN
PRINT "Even"
ELSE
PRINT "Odd"
END IF
```

SELECT CASE, CASE ELSE, END SELECT

Syntax
```
SELECT CASE varName
CASE varExp [ { , varEXP } ] instruc-
tion
[ { CASE varExp [ { , varEXP } ]
instruction } ]
CASE ELSE instruction
END SELECT
```

Parameters
`varName` is a valid variable name,
`varEXP` are expressions of the same type as `varName`,
`instruction` are valid RAPID! commands.

Description
Performs one of several blocks of commands, depending on the value of `varName`.
If `varName` is identical with any one of the `varExp` expressions, the commands following that `CASE` statement will be performed until the subsequent `CASE` or the `END SELECT` command is reached. The program will then resume with the command following the `END SELECT` statement.
If `varName` is identical with several `varExp`, only the first one found will be executed.
Several `varExp` may be used in the same `CASE` command; they need to be separated by commas.

Examples
```
SELECT CASE Day
CASE 1: PRINT "MONDAY"
CASE 2: PRINT "TUESDAY"
CASE 3: PRINT "WEDNESDAY"
CASE 4: PRINT "THURSDAY"
CASE 5: PRINT "FRIDAY"
CASE 6, 7: PRINT "WEEKEND"
CASE ELSE
PRINT "Data out of range"
END
END SELECT
```

```
SELECT CASE Day$
CASE "MONDAY","TUESDAY","WEDNESDAY",
"THURSDAY,"FRIDAY"
PRINT "Got to work!"
CASE "WEEKEND" : PRINT "I'm off for
the seaside!"
END SELECT
```

Branch commands

With the help of these commands, it is easy to branch program execution, e.g. depending on conditions or user entries.

Command	Short description
ERROR	Creates an error with a specified error number.
GOSUB ... RETURN	Branches to the specified label and returns at the next RETURN command.
GOTO	Branches to the specified label.
ON ERROR	Branches to the specified label, if an error occurs during program execution.

ERROR

Syntax   ERROR errNo

Parameters   errNo is an integer number.

Description   Creates a runtime error with the specified error number. This command enables you to create your own error codes.
Note: For the built-in error codes, please refer to the table of Runtime errors.

Examples   The following example defines an error code based on a simple condition and gives a first idea of error handling.
```
IF (frequ > 2 GHz) THEN ERROR 57
...
SELECT CASE err
case 57 : Print "Input out of range!"
END CASE
```

GOSUB ... RETURN

Syntax   GOSUB targetLabel
                    [instructions]
                    RETURN

Parameters   targetLabel is a label.

Description	Branches the program to the symbolic label and continues program execution with the commands following that label.
	As soon as the RETURN command is encountered, program execution will come back to the command following the initial GOSUB command.
| **Example** | ```
IF Day$="Friday" THEN GOSUB fishDay
ELSE GOSUB meatDay
...
fishDay
PRINT "Today we eat fish!"
RETURN
...
meatDay: PRINT "Today we will have a
steak!"
RETURN
``` |

GOTO

Syntax	`GOTO targetLabel`
Parameters	`targetLabel` is a label.
Description	Branches to the symbolic label and continues program execution with the commands following that label.
| **Examples** | ```
...
IF i>10 THEN GOTO loopOver
...
loopOver: PRINT "Ten steps later..."
``` |

ON ERROR, RESUME

Syntax	`ON ERROR { GOTO {targetLabel	0} |` `RESUME NEXT }` `...` `RESUME [NEXT	targetLabel]`
Parameters	`targetLabel` is a label.

Description   If an error occurs during program execution, the program would normally be terminated.

The `ON ERROR` command structure gives you the chance to trap the error, handle it, and then continue the program without a break.

`ON ERROR GOTO targetLabel` branches to the specified label in case an error occurs.

`ON ERROR GOTO 0` deactivates the error trapping. Any subsequent error will stop the program.

`RESUME` continues the program with the command that caused the error.

`RESUME targetLabel` continues the program at the specified label.

`RESUME NEXT` continues the program with the command following the one, that caused the error.

Note: If an error occurs within the error handling, the program will be terminated.

There are several built-in functions available for error handling.

Examples   In the first example, the 4400 prints an error message, giving the error code, the line in which the error occurred and the name of the related RAPID! program file.

```
ON ERROR GOTO errorHandler
...
errorHandler
PRINT "Error "; ERR; " in line "; ERL;
" in file "; ERF$; " Resuming..."
RESUME NEXT
```

In the following example, a user-specific error code is generated first. Then a `CASE SELECT` command is used to print the correct error message.

```
ON ERROR GOTO errorHandler
...
LET measQ = measNum / measRes
IF (measRes<0) THEN ERROR 57
...
errorHandler: SELECT CASE ERR
CASE 104: PRINT "Division by zero!
Restart measurement!"
CASE 57: PRINT "Measurement result
negative! Restart measurement!"
CASE ELSE
PRINT "Unknown error. Program halts."
END
END CASE
RESUME NEXT
```

**Commands for input/
output handling**

These commands enable special input/output tasks related to the 4400's disk drive, hard disk and communication ports.
Frequently, input/output operations are abbreviated with I/O.

Command	Short description
CLOSE	Closes a previously opened file.
INPUT	Reads data from an open file or from an open port.
OPEN	Opens a file or a port and allows for reading or writing data.
PRINT or OUTPUT	Writes data to an open file or port.

CLOSE

Syntax

```
CLOSE [fileNo]
```

Parameters

`fileNo` is a numeric expression, representing a valid file number. `fileNo` must result in a positive integer.
Software people tend to call `fileNo` the 'file handle'.

Description

Closes the file with the specified `fileNo`.
After the file has been closed, `fileNo` is no longer related to the initial file and may be used again later to open a completely different file.
If `fileNo` is omitted, then all currently open files will be closed.
As long as a file is closed, it is not accessible for I/O operations unless it is opened again, using the OPEN command (please see below for details).

Example

The following example program reads a text file and stores the data read from the file in a string array.

```
LET i=0
OPEN "STRINGS.TXT" FOR INPUT AS #1
DO WHILE NOT EOF(1)
LET i=i+1
INPUT #1, theStrgs$(i)
LOOP
CLOSE #1
```

INPUT

Syntax

```
INPUT #fileNo , variable
```

Parameters

`fileNo` is a numeric expression resulting in a valid file handle (a positive integer).
`variable` is a numeric or string variable.

Description
Reads data from a previously opened file or communication port with the specified file handle into a variable. `INPUT` will always read a complete line and then increment the line counter. This means that a second `INPUT` command with the same file handle will read the second line of the file.

Notes:

– Before `INPUT` may read from a file, it needs to be opened for input first (please see the description of the `OPEN` command and the `INPUT` mode in this subsection). If you try to read from a file that was not opened before, a runtime error will occur.

– It is recommended that you check if the end of the file was reached before you issue a subsequent `INPUT` command: if `INPUT` tries to read after the end of the file was reached, a runtime error will occur.

– With the help of the `INPUT` command, you may also prompt the user to enter data during runtime of the program. Please see subsection "Screen commands" on page 290 for reference.

– `INPUT` also reads data from the SCPI system. Please refer to section SCPI and RAPID! for details.

Example
The following program reads a text file and stores the data read from the file in a string array.

```
LET i=0
OPEN "STRINGS.TXT" FOR INPUT AS #1
DO WHILE NOT EOF(1)
LET i=i+1
INPUT #1, theStrgs$(i)
LOOP
CLOSE #1
```

`OPEN` **(on files)**

Syntax
```
OPEN fileName$ FOR openMode AS
#fileNo
```

Parameters
`fileName$` is a string expression containing a valid file name including an optional path or the specification for a communication port of the 4400 (please see below for details).
`openMode := {APPEND | INPUT | OUTPUT}`
`fileNo` is a numeric expression resulting in a positive integer (the so-called file handle).

Description	Opens a file or a communication port for input or output operations. As long as a file has not been opened, I/O operations can not be performed and a runtime error will occur.

`filename$` is a string, containing either a valid file name or a specification for a communication port of the 4400 (please see below for details).

`openMode` may either be `INPUT`, `OUTPUT` or `APPEND`. When `openMode` is `INPUT`, then the file can only be read from. If the file does not exist, a runtime error will occur.

In case, `openMode` is `OUTPUT`, then it is only possible to store data to that file. If the file does not exist, it will be created anew. Should the file already exist, then it will be **overwritten without warning**.

`openMode` `APPEND` is similar to `OUTPUT`. However, the file will not be overwritten with new data. Instead, new data will be added at the end of the existing file. Should there be no file, it will be created anew.

Notes:

– When a file is already open, a further `OPEN` command may open the same file as long as the mode is identical. If the mode is not identical (e.g. a file was opened for output previously and shall now be opened for input), a runtime error will occur.

– `OPEN` may also be used to open a path to the SCPI system. See section SCPI and RAPID! for details.

Examples

This example writes a string to a text file.

```
LET File=FREEFILE
OPEN "STRINGS.TXT" FOR OUTPUT AS
#File
PRINT #File, "Save me"
CLOSE #File
```

`OPEN` (on communication ports)

Syntax

```
OPEN fileName$ AS #fileNo
```

Parameters

`fileName$` is a string expression containing a specification for a communication port on the 4400 or a valid file name (please see above for details).
fileNo is a numeric expression resulting in a positive integer (the so-called file handle).

Description   Opens a communication port for input or output opera-
tions. As long as a port has not been opened, I/O opera-
tions can not be performed and a runtime error will
occur.

`filename$` is a string, containing a specification for a
communication port of the 4400 or a valid file name
(please see above for details).

Valid ports are GPIB0 (for the GPIB/IEEE-Bus), SCPI,
COM1 and COM2 (the two serial interfaces of the 4400,
please see "Rear panel" on page 8 for reference).
A port name always needs to be followed by a colon (:).
After the colon, you may specify parameters for the
interface and the communication.

Please note that the values of parameters must be posi-
tive integers if not specified otherwise.

Parameters for GPIB:

`addr=` this parameter specifies the primary GPIB bus
address. The valid range is 1…32.
`sec=` states the secondary GPIB bus address. The valid
range is 0…64.
`time=` with the help of this parameter, you may set a
timeout in milliseconds.
`eoi=` states, whether the hardware EOI will be used
(parameter set to 1) or not (parameter set to 0).
`srq=` a '1' states that the hardware 'Service Request' will
be used, while a '0' indicates that it will not be used.
`end=` specifies the end byte (usually ASCII 10 or ASCII
13). The valid range is 0…255.
Parameters for SCPI:

There are no parameters for SCPI.

Parameters for COM1/COM2:

`baud=` this parameter specifies the baud rate of the
serial interface. It is highly recommended to always spec-
ify a baud rate when working with the serial ports to
keep them in a defined state. The following values may
be entered: 110, 300, 600, 1200, 2400, 4800, 9600,
19200, 38400, 57600, 76800, 115200.

`Bits=` states the number of data bits used. The valid
range is 5…8.
`stop=` specifies the number of stop bits to be used. The
entry range is 1│2.
`parity=` states, whether the parity check bit will be
used. Value '0' switches the use of a parity bit off. '1'
indicates odd parity and '2' even parity.

`xonoff=` defines whether the software handshake will be used. A '0' switches the software handshake off, while '1' turns it on. Software handshake means that the receiver (RAPID! program) may halt the transmission by sending ASCII 19 (Ctrl-S) to the transmitter. A Ctrl-Q (ASCII 17) sent to the transmitter will continue transmission again. Default value is '0' (no software handshake).

`rts=` defines whether the hardware handshake will be used. A '0' switches the hardware handshake off, while '1' turns it on. Hardware handshake means that the receiver (RAPID! program) will only allow the transmitter to send while it is expecting data in an `INPUT` command. Physically, pin RTS will be kept to 'low' and only turn to 'high' when the RAPID! program expects input data. Default value is '0' (no hardware handshake).

`cts=` states, whether the signal on pin CTS will be ignored (value = '1') or listend to (value = '0'). If `cts=0`, the transmitter is only allowed to send data as long as the signal on pin CTS is active (i.e. 'high'). In case CTS is to be ignored `cts=1`, the receiver (RAPID! program) will always wait until all characters have been transmitted (please see parameter `wait=`). Default value is '0' (CTS signal will be followed).

`time=` with the help of this parameter, you may set a timeout in seconds. When the timeout expires, an empty string will be delivered to the RAPID! program as input. Default is 600 seconds (5 minutes).

`end=` specifies the end byte (usually ASCII 10 or ASCII 13). The valid range is 0...255. If an end byte was specified, data will be read in until the specified end byte is recognized. If no end byte is specified (`end=0`), only the currently available bytes will be read in. Default is `end=0`.

`flush=` defines the use of the port input buffer. In case `flush=0`, all characters will be read in, even when they occurred at the port long before the port was opened with the `OPEN` command. `flush=1` however clears the input buffer in the moment, the `OPEN` command for that port occurs in the program run. Default is `flush=0` (all data stored in input buffer will be kept).

`wait=` specifies whether the program is to wait until all characters have been transmitted over the port, or not. `wait=0` will continue the program after the transmission of data has been initialized. `wait=1` will halt the program until all characters have been transmitted. Default is `wait=0`.

Notes:

The SCPI port may only be opened once. If you try to open the previously opened SCPI port, a runtime error will occur.

All other ports may be opened several times with different parameters as long as the file handles are different.

Examples

This example opens the GPIB0 port.

```
gpib=FREEFILE
OPEN "GPIB0:addr=25,sec=0,time=10,
eoi=1,srq=1,end=13" AS #gpib
PRINT #gpib, "Output to GPIB"
INPUT #gpib, ingpib$
CLOSE #gpib
```

The program segment below opens a serial interface.

```
com2=FREEFILE
open-
com$="COM2:baud=9600,bits=8,stop=1,
parity=1,xonoff=0"
opencom$=opencom$ +
"time=10,end=10,flush=1,wait=1"
open opencom$ as #com2
PRINT #com2, "Output to serial port
2"
INPUT #com2, incom2$
CLOSE #com2
```

`PRINT `**`or`**` OUTPUT`

Syntax

```
PRINT #fileNo [USING formatstring ,] exp
[ [TAB[n];] [BYTE(m)] ] [{ ,|; exp} ]
```
or
```
OUTPUT #fileNo [USING formatstring ,] exp
[ [TAB[n];] [BYTE(m)] ] [{ ,|; exp} ]
```

Parameters

`exp` is a numeric or string expression.
fileNo is a numeric expression resulting in a positive integer (the so-called file handle).
`formatstring` specifies the output format to be used, e.g. the number of decimal places.
`n` is an optional parameter giving the distance between the previous and this tabulator position.
`m` is the decimal representation of the ASCII character to be printed.

Description Writes the specified expression(s) to a previously opened file, using the specified file handle.

With the help of this command, several expressions may be printed in the same line. When the separator is a comma (,), the subsequent expression will be written at the next tabulator position.

When the separator is a semicolon (;), the subsequent expression will be written directly behind the previous one.

Notes:

- Before PRINT may output to a file, it needs to be opened for output or appending first (please see the description of the OPEN command and the OUTPUT and APPEND modes in this subsection). If you try to write to a file that was not opened before, a runtime error will occur.

- With the help of the PRINT command, you may also print data to the 4400's screen. Please refer to subsection "Screen commands" on page 290 for details.

- PRINT also reads data from the SCPI system. Please check with section "SCPI and RAPID!" on page 347 for details.

Examples This example writes a string to a text file.

LET File=FREEFILE

OPEN "STRINGS.TXT" FOR OUTPUT AS #File

PRINT #File, "Save me"

CLOSE #File

Please compare the use of the PRINT command in this context with the third example for the PRINT command in the Screen commands description, too.

Functions

The functions of the RAPID! environment are subdivided into five groups:

– Numeric functions. They convert numeric values into different systems or return numerical values. Typical examples are BIN, BIN$, CINT, OCT, OCT$, HEX, HEX$, VAL or VAL$.

– String functions operate on strings or return string values. Typical examples are ASC, CHR$, INSTR, LEN, LEFT$, MID$, RIGHT$ or SPACE$.

– I/O functions provide useful information concerning input and/or output operations. Typical examples are DATE$, CLOCK, EOF or FREEFILE.

– Functions for error handling return values or strings related to errors. ERR, ERL and ERF$ belong to this group.

– Mathematical functions support program-internal calculations with the mathematical functions SQRT, SIN, COS or TAN just to name a few.

Numeric functions

This subsection describes functions that convert numbers or perform numeric operations.

Note: In case uninitialized variables are used as arguments, the returned values will usually be `"0"` or `0`.

Function	Short description
BIN	Converts a binary representation of a numeric expression into its decimal equivalent.
BIN$	Transforms a numeric expression into its binary representation.
CINT	Rounds a numeric expression to an integer, using commercial rounding.
HEX	Converts a hexadecimal representation of a numeric expression into its decimal equivalent.
HEX$	Transforms a numeric expression into its hexadecimal representation.
OCT	Converts an octal representation of a numeric expression into its decimal equivalent.
OCT$	Transforms a numeric expression into its octal representation.
VAL	Converts a string expression into a numeric expression.
VAL$	Converts a numeric expression into a string expression.

BIN

Syntax	`BIN(string$)`
Parameters	`string$` is a string expression, containing a valid binary representation of an integer number.
Description	Returns the decimal representation of a binary expression contained in `string$`. The value returned will be an integer.
Examples	`LET i=BIN("101")` assigns '5' to variable `i`.

BIN$

Syntax	`BIN$(aNumber [, digits ])`
Parameters	`aNumber` is a valid numeric expression, `digits` is an expression, resulting in a positive integer.

Description		Returns a string containing the binary representation of `aNumber`. In case, `aNumber` is not an integer, the decimals will be truncated (that means no rounding will take place).
		When `digits` is stated, the string will be filled up with zeros (0) to the left.
Examples		`LET b$=BIN$(126, 8)` assigns `"01111110"` to variable `b$`.

CINT

Syntax	`CINT(aNumber)`
Parameters	`aNumber` is a valid numeric expression.
Description	Converts a numeric expression to the related integer figure, using commercial rounding.
Examples	`i = CINT(1.49)` assigns '1' to variable `i`.
	`j = CINT(1.50)` assigns '2' to variable `j`.

HEX

Syntax	`HEX(string$)`
Parameters	`string$` is a string expression, containing a valid hexadecimal representation of an integer number.
Description	Returns the decimal representation of a hexadecimal expression contained in `string$`.
	The value returned will be an integer number.
Examples	`LET i=HEX("FF")` assigns '255' to variable `i`.

HEX$

Syntax	`HEX$(aNumber [, digits ])`
Parameters	`aNumber` is a valid numeric expression,
	`digits` is a numeric expression, resulting in a positive integer.
Description	Returns a string containing the hexadecimal representation of `aNumber`. In case, `aNumber` is not an integer, the decimals will be truncated (that means no rounding will take place).
	When `digits` is stated, the string will be filled up with zeros (0) to the left.
Examples	`LET a$=HEX$(254, 4)` assigns `"00FE"` to variable `a$`.

OCT

Syntax	`OCT(string$)`
Parameters	`string$` is a string expression, containing a valid octal representation of an integer number.
Description	Returns the decimal representation of an octal expression contained in `string$`. The value returned will be an integer number.
Examples	`LET i=OCT("15")` assigns '13' to variable `i`.

OCT$

Syntax	`OCT$(aNumber [, digits ])`
Parameters	`aNumber` is a valid numeric expression, `digits` is a numeric expression, resulting in a positive integer.
Description	Returns a string containing the octal representation of `aNumber`. In case, `aNumber` is not an integer, the decimals will be truncated (that means no rounding will take place). When `digits` is stated, the string will be filled up with zeros (0) to the left.
Examples	`LET a$=OCT$(254, 6)` assigns `"000376"` to variable `a$`.

VAL

Syntax	`VAL(string$)`
Parameters	`string$` is a string expression, containing a valid numeric representation of a decimal figure like "-1.4" or "3E+5"
Description	Converts a decimal number contained in `string$` to that decimal number. If the string does not contain a valid decimal number, '0' will be returned.

Examples	`i = VAL("1.234")` assigns '1.234' to variable `i`
	To avoid runtime errors due to typing errors with an `INPUT` command, first a string is read in that is later converted into the decimal representation of the figure entered.
	`INPUT "Please enter channel number:" _`
	`, channelno$`
	`channelno = VAL(channelno$)`

VAL$

Syntax	VAL$(aNumber [, digits])
Parameters	aNumber is a numeric expression. digits is a numeric expression, resulting in a positive integer.
Description	Returns a string containing the numeric representation of aNumber. The number of decimal places may be specified using the optional digits command element. In case aNumber uses more decimal places than specified with digits, the additional decimal places will be truncated (i.e. there will be no rounding). If digits is omitted, six decimal places will be used.
Examples	VAL$(123) returns the string "123.000000". VAL$(123.456789, 2) returns the string "123.45". VAL$(123.456789, 0) returns the string "123".

String functions

This subsection describes functions that operate on strings.

Note: In case uninitialized variables are used as arguments, the returned values will usually be empty strings or '0'.

Function	Short description
ASC	Returns the ASCII code of the first character in a string.
CHR$	Returns the character representing the ASCII value of an integer number as a string.
INSTR	Searches for the occurrence of a specific sequence of characters within a string.
LEN	Returns the number of characters included in a string.
LEFT$	Reads a specific part of a string, starting from its beginning.
MID$	Reads a specific part of a string.
RIGHT$	Reads a specific part of a string, starting from its end.
SPACE$	Returns a string, filled with a specific number of blanks.

ASC

Syntax	ASC(aString$)
Parameters	aString$ is a valid string expression.
Description	Returns the ASCII code of aString$'s first character. If the string is empty, '0' will be returned.

Examples ASC("A") will return '65'.

CHR$

Syntax	CHR$(aNumber)
Parameters	aNumber is a valid numeric expression.

Description Returns the character corresponding to the ASCII code of aNumber as a string.
If aNumber is outside the range 0...255, then its least significant byte will be used (i.e. a modulo 256 operation will be performed on aNumber).
Should aNumber have decimal places, then those will be truncated (this means that there will be no rounding).
If this command is used on a variable that has not been initialized before, '0' will be returned.

Examples a$=CHR$(65) assigns "A" to variable a$.

INSTR

Syntax	INSTR([startPos,] aString$, search-String$)
Parameters	startPos is a numeric expression, interpreted as positive integer. aString$ and searchString$ are string expressions.

Description Searches aString$ for the first occurrence of search-String$.
The search will start at the position given by startPos or, if omitted, at the beginning of aString$.
Return value is the position, where searchString$ begins.
If searchString$ is not found within aString$, '0' will be returned.
Should startPos have decimal places, then those will be truncated (this means that there will be no rounding).

Examples a$ = "international or national"
i = INSTR(a$, "national")
j = INST(8, a$, "national")
In this example, variable i will be set to '6', while variable j will be set to '18'.

LEN

Syntax	LEN(aString$)
Parameters	aString$ is a string expression.

Description Returns the number of characters contained in aString$.

| | Examples | LEN("A short one") will return '11'. |

LEFT$

	Syntax	LEFT$(aString$, chars)
	Parameters	aString$ is a string expression, chars is a numeric expression, interpreted as positive integer.
	Description	Returns the first chars characters of aString$, or the whole string if chars is greater than the amount of characters contained in aString$. Should chars have decimal places, then those will be truncated (this means that there will be no rounding).
	Examples	a$="Good morning" b$=LEFT$(a$,4) Will assign the string "Good" to b$.

MID$

	Syntax	MID$(aString$, startPos, chars)
	Parameters	aString$ is a string expression, startPos and chars are numeric expressions, interpreted as positive integers.
	Description	Returns chars characters of aString$, starting at position startPos. If startPos+chars is greater than the total amount of characters contained in aString$, the entire string, starting from startPos will be returned. If chars is omitted, the entire string, starting from startPos will be returned. If startPos is greater than the total amount of characters contained in aString$ (i.e. startPos points past the end of aString$), then an empty string will be returned. Should startPos or chars have decimal places, then those will be truncated (this means that there will be no rounding).
	Examples	MID$("This is a test", 6, 4) will return the string "is a".

RIGHT$

| | Syntax | RIGHT$(aString$, chars) |
| | Parameters | aString$ is a string expression, chars is a numeric expression, interpreted as positive integer. |

Description	Returns the last `chars` characters of `aString$`, or the whole string if `chars` is greater than the amount of characters contained in `aString$`. Should `chars` have decimal places, then those will be truncated (this means that there will be no rounding).

Examples	`a$="Good morning"`
	 `c$=RIGHT$(a$,7)`
	 Will assign the string "morning" to `c$`.

SPACE$

Syntax	`SPACE$(chars)`
Parameters	`chars` is a numeric expression, interpreted as positive integer.
Description	Returns a string containing a total amount of `chars` blanks.
Examples	`d$ = "Long"+SPACE$(1)+"distance"` will assign the string "Long distance" to `d$`.

I/O functions

The 4400's RAPID! environment provides a large number of I/O functions:

Function	Short description
`CHDIR`	Changes the current directory.
`CLOCK`	Returns the time since the system was last started.
`CURDIR$`	Returns the currently selected directory.
`DATE$`	Returns a string, containing the system date.
`DIR$`	Function to read out the entries in a directory.
`EOF`	Checks whether the end of a file was reached.
`EVENTWAIT, EVENTSTATUS`	Waits for an event and returns the event code.
`FREEFILE`	Returns the subsequent free file handle.
`INKEY`	Checks whether a softkey was pushed and returns its code.
`INKEYWAIT`	Halts the program until a softkey is pushed.
`KILL`	Deletes a file.
`MKDIR`	Creates a new directory.
`NAME`	Renames an existing file.
`RMDIR`	Removes a directory.

SHELL   Calls the operating system shell in order to perform operating system commands (e.g. so-called pipes).

TIME$ Returns a string containing the system time.

CHDIR

Syntax	`CHDIR path$`
Parameters	`path$` is a valid string expression.
Description	Changes the current directory to the directory specified in `path$`. `path$` may either contain the absolute or relative path. If `path$` is not a valid path, the current directory will be kept and a runtime error will occur.
Example	CHDIR = "/usr/new"

CLOCK

Syntax	`CLOCK`
Parameters	There are no parameters.
Description	Returns the time since the system was started last. The value returned is a floating-point double value, expressing the time expired since the last start of the system in seconds.
Example	`t = CLOCK` Assigns the time since the system was started last to variable t (t would then be set to e.g. '3724.23' (seconds)).

CURDIR$

Syntax	`CURDIR$`
Parameters	There are no parameters.
Description	Returns the currently selected directory.
Example	`PRINT "Current directory:";CURDIR$`

DATE$

Syntax	`DATE$`
Parameters	There are no parameters.

Description	Returns a string containing the system date in the following format: "mmm-dd-jjjj", where "mmm" represent a three-letter abbreviation of the month, "dd" stand for the integer, giving the day of the month and "jjjj" represent the current system year in four-digit code.
Examples	DATE$ would have returned "Jan-03-2001" on the day this subsection was created.

DIR$

Syntax	DIR$ [(path$)]
Parameters	path$ is a valid string expression.
Description	Returns an entry in the directory specified with the optional command element path$. DIR$ will read the first entry in the specified directory only. Repeated DIR$ function calls (without the path command element) will read out the subsequent entries of the currently selected directory one by one. If there are no more entries, an empty string will be returned. path$ may either contain the absolute or relative path. If path$ is not a valid path, the returned DIR$ will be empty and a runtime error will occur.
Example	The following example will print a list of all entries in the directory specified.

```
PRINT "Directory entries:";DIR$("/
home")
DO    d$ = DIR$
   IF d$ = "" THEN EXIT DO
   PRINT d$
LOOP
```

EOF

Syntax	EOF(fileHandle)
Parameters	fileHandle is a numeric expression, interpreted as positive integer.
Description	Checks whether the end of the file, specified by file-Handle, has been reached. If so, EOF will return 1 (or true), otherwise 0 (or false). fileHandle needs to be a currently valid file handle, and the file needs to be opened first, before EOF may be used (otherwise, a runtime error will occur). **Note:** When you try to read from a file with the INPUT command after the end of the file was reached, a runtime error will occur. To avoid that, it is highly recommended to check EOF, before you issue an INPUT command.

Example   The following example program reads a text file and
stores the data read from the file in a string array.

```
LET i=0
OPEN "STRINGS.TXT" FOR INPUT AS #1
DO WHILE NOT EOF(1)
    LET i=i+1
    INPUT #1, theStrgs$(i)
LOOP
CLOSE #1
```

EVENTWAIT, EVENTSTATUS

Syntax   `EVENTWAIT(timeOut)`
`EVENTSTATUS`

Parameters   `timeOut` is a numeric expression, interpreted as positive
integer.

Description   `EVENTWAIT`
Waits for an event and returns the event code.
Events may be created by various devices or tasks (e.g.
reception of a string on the GPIB bus).
`timeOut` specifies a wait time in milliseconds. During
the wait time, the program will be halted and will wait
for the event to occur. If the expected event does not
occur, a timeout will be generated.
`EVENTWAIT` may be used to react on any of the follow-
ing events:

KEY push of a softkey. In this case, the file handle
returned will be '0'.

TIME–OUT means that there was no device or task creat-
ing an event; time simply expired without anything hap-
pening. In this case, the file handle returned will be '-1'.

GPIB an event on the GPIB bus (e.g. data have been
applied and are ready to be read).

SCPI an event on the SCPI interface (e.g. a measurement
value is ready to be handed over).

`EVENTSTATUS`
In case there is an event created by the GPIB bus,
`EVENTSTATUS` may be used to read the status byte.

Example The following example labels the softkeys, opens several communication ports and then waits 5 seconds for an event to occur.
Depending on the device or task creating the event, event handling will take place.

```
SOFTKEYS "OK",,,,,,,"EXIT"
OPEN "GPIB0:..." AS #1
scpi = FREEFILE
OPEN "SCPI:" AS #scpi
DO
SELECT CASE EVENTWAIT(5000)
   CASE -1: PRINT "Time out"
   CASE 0
      IF INKEY = 7 THEN END
      PRINT "Softkey 'OK' was pushed."
   CASE 1
      INPUT #1, gpib$
      PRINT "Received on GPIB:",
gpib$,EVENTSTATUS
CASE SCPI
      INPUT #scpi, scpi$
      PRINT "Received from SCPI:", scpi$
   CASE ELSE
      PRINT "Any other event occurred!"
END SELECT
LOOP
```

FREEFILE

Syntax `FREEFILE`

Parameters There are no parameters.

Description Returns the subsequent available file handle as a positive integer.
The returned file handle may then be used by an OPEN (on files) command.
`FREEFILE` helps to avoid conflicts with file handles.

Example This example writes a string to a text file.
```
LET File=FREEFILE
OPEN "STRINGS.TXT" FOR OUTPUT AS
#File
PRINT #File, "Save me"
CLOSE #File
```

INKEY

Syntax `INKEY`

Parameters There are no parameters.

Description	Checks whether a softkey was pushed and returns the number of the softkey. The softkeys on the 4400 are numbered from '1' to '7', starting from the left. If no softkey was pushed, INKEY will return '0'. In contrary to the INKEYWAIT command, INKEY does not halt the program. **Note:** Please be sure that there have been labels assigned to the softkeys with the SOFTKEYS command, **before** you use INKEY to read out the softkey pushed.
| **Example** | ```
SOFTKEYS "Accept", "", "", "", "", "",
"Exit"
DO
    IF INKEY = 7 THEN GOTO endProg
LOOP
``` |

INKEYWAIT

Syntax	`INKEYWAIT`
Parameters	INKEYWAIT does not have any parameters.
Description	Halts program execution until the user pushes a softkey. **Notes:** – If you want to determine the softkey pushed without halting the program, choose the INKEY function instead. – Please assign labels to the softkeys using the SOFTKEYS command, before using INKEYWAIT. Otherwise, the program may wait for eternity.
| **Example** | ```
SOFTKEYS "OK", "", "", "", "", "",
"EXIT"
IF INKEYWAIT = 7 THEN END
``` |

KILL

Syntax	`KILL fileName$`
Parameters	fileName$ is a valid string expression.
Description	Deletes the file specified in fileName$ in the currently set directory without prompting. fileName$ may also contain a path. If fileName$ is not a valid file name or if fileName$ is currently open, a runtime error will occur.
Examples	`KILL "/usr/new/file.txt"`

MKDIR

Syntax	`MKDIR path$`

Parameters	`path$` is a valid string expression.
Description	Creates a new directory with the `path$` specified. `path$` may either be a relative or absolute directory name. If `path$` is not a valid directory name, a runtime error will occur.
Examples	`MKDIR "/usr/new"`

NAME

Syntax	`NAME oldName$ AS newName$`
Parameters	`oldName$` and `newName$` are both valid string expression.
Description	Changes the name of the file specified with `oldName$` to `newName$`. Both `oldName$` and `newName$` have to be valid file names. If `oldName$` is currently open, a runtime error will occur.
Example	`NAME "/usr/new/file.txt" AS "/usr/new/file_2.txt"`

RMDIR

Syntax	`RMDIR path$`
Parameters	`path$` is a valid string expression.
Description	Deletes a directory with the `path$` specified without prompting. `path$` may be either a relative or an absolute directory name. If `path$` is not a valid directory name or if the directory is not empty, a runtime error will occur.
Example	`RMDIR "/usr/new"`

SHELL

Syntax	`SHELL parameter$`
Parameters	`parameter$` is a valid string expression.

Description	Calls a command of the operating system, specified in `parameter$`. `parameter$` may contain all legal commands of the operating system including the pipe ('>') symbol. Thus, output generated by a command of the operating system may be routed to a file and read in by a RAPID! program. If `parameter$` does not contain a valid command of the operating system, a runtime error will occur.
Examples	The following one-line program reads the file list of all files contained in a specified directory and stores them in a text file. `SHELL "ls /usr/new > /usr/new/ls.txt"`

TIME$

Syntax	`TIME$`
Parameters	There are no parameters.
Description	Returns a string containing the system time in the following format: "hh:mm:ss", where "hh" is a positive integer, giving the current system hour (in 24 h format), "mm" give the current minutes and "ss" represent the current seconds.
Examples	`TIME$` would have returned "10:29:37" at the time this table was created.

Functions for error handling

For program-internal error handling, the following functions have been implemented in RAPID!

Function	Short description
ERR	Returns the error code of the most recent error.
ERL	Gives the program line, in which the last error occurred.
ERF$	Returns the name of program file, in which the last error occurred.

ERR

Syntax	`ERR`
Parameters	There are no parameters.
Description	Returns the error code of the latest error that occurred. For the meaning of the error codes, please see "Runtime errors" on page 337 for reference.

Example In this example, the 4400 prints an error message, giving the error code, the line in which the error occurred and the name of the related RAPID! program file.

```
ON ERROR GOTO errorHandler
...
errorHandler
    PRINT "Error "; ERR; " in line ";
ERL;" in file ";ERF$
RESUME NEXT
```

ERL

Syntax `ERL`

Parameters There are no parameters.

Description Returns the number of the program line, in which the last error occurred.

Example Please see example with `ERR` function above.

ERF$

Syntax `ERF$`

Parameters There are no parameters.

Description Returns the name of the RAPID! program file, in which the last error occurred.

Examples Please see example with `ERR` function above.

Mathematical functions

The RAPID! environment of the 4400 provides most of the important mathematical functions:

Function	Short description
ABS	Returns the absolute value of a numeric expression.
LOG, LGT	Calculates the natural logarithm and the common logarithm, repectively.
RND, RANDOM-IZE	Functions for generating random numbers.
SIN, COS, TAN, ATAN	The most important trigonometric functions.
SQRT	Calculates the square root of a numeric expression.

| | SGN | Returns the sign of a numeric expression. |

ABS

	Syntax	`ABS(numExp)`
	Parameters	`numExp` is a valid numeric expression.
	Description	Returns the absolute value of `numExp`. If `numExp` is an unassigned variable, a '0' will be returned.
	Examples	`t = ABS(-1)` Assigns '+1' to variable `t`.

LOG, LGT

	Syntax	`LOG(numExp)` `LGT(numExp)`
	Parameters	`numExp` is a valid numeric expression.
	Description	`LOG` returns the natural logarithm of `numExp` while `LGT` returns the common logarithm of `numExp`.

	Examples	`i = LOG(1000)`
		`j = LGT(1000)`
		Variable `i` will be set to '6.907755' while `j` will be set to '3'.

RND, RANDOMIZE

	Syntax	`RND` `RANDOMIZE(numExp)`
	Parameters	`numExp` is a valid numeric expression, interpreted as positive integer.
	Description	`RND` generates a random double floating-point number between '0' and '1'. `RANDOMIZE` is used to reinitialize the random generator. **Note:** `RANDOMIZE(x)` will always lead to the same sequence of random numbers.

	Examples	The following example program reinitializes the random generator and then prints ten random numbers.
		`RANDOMIZE 321`
		`FOR i = 1 TO 10`
		`    PRINT RND`
		`NEXT i`

SIN, COS, TAN, ATAN

Syntax	`SIN(numExp)`
	`COS(numExp)`
	`TAN(numExp)`
	`ATAN(numExp)`

Parameters	`numExp` is a valid numeric expression.

Description

`SIN(numExp)`

Calculates the sine of `numExp` and delivers back a double floating-point number.

`numExp` is interpreted as the radiant expression of the angle (SIN(3.14...) = 0).

`COS(numExp)`

Calculates the cosine of `numExp` and delivers back a double floating-point number.

`numExp` is interpreted as the radiant expression of the angle (COS(3.14...) = –1).

`TAN(numExp)`

Calculates the tangent of `numExp` and delivers back a double floating-point number.

`numExp` is interpreted as the radiant expression of the angle (TAN(3.14...) = 0).

`ATAN(numExp)`

Calculates the arcus tangent of `numExp` and delivers back a double floating-point number.

`numExp` is interpreted as the radiant expression of the angle (ATAN(3.14...) = 1.262627).

Examples   Please see examples nested in the description above.

SQRT

Syntax	`SQRT(numExp)`
Parameters	`numExp` is a valid numeric expression.
Description	Returns the square root of `numExp` as a double floating-point number.
Examples	`i = SQRT(25)` Assigns '5' to variable `i`.

SGN

Syntax	SGN(numExp)
Parameters	`numExp` is a valid numeric expression.
Description	Returns the sign of `numExp`. If `numExp` is equal to or greater than '0', `SGN` will return '+1'. If `numExp` is less than '0', `SGN` will return '-1'.

Examples
```
i = SGN(234)
j = SGN(-0.023)
```

Variable i will be set to '+1', while variable j will be set to '-1'.

Tables

While working with RAPID!, it is quite helpful to keep the following tables at hand:

– List of all RAPID! commands and functions in alphabetic order.
– Table of all Syntax errors, giving also their meaning and hints how to trace the problem.
– Table of all Runtime errors, explaining their meaning and giving hints of how to solve the matter.

RAPID! commands and functions

In this table, you will find all RAPID! commands and functions in alphabetic order for a quick overview.

Command or function	Short description
ABS	Returns the absolute value of a numeric expression.
ASC	Returns the ASCII code of the first character in a string.
ATAN	Calculates the arcus tangent of a numeric expression.
BIN	Converts a binary representation of a numeric expression into its decimal equivalent.
BIN$	Transforms a numeric expression into its binary representation.
CHAIN	Loads and executes a different RAPID! program.
CHDIR	Changes the current directory.
CHR$	Returns the character representing the ASCII value of an integer number as a string.
CINT	Rounds a numeric expression to an integer, using commercial rounding.
CLOCK	Returns the time since the last start of the system.
CLOSE	Closes a previously opened file.
CLS	Clears the screen.

COS	Returns the cosine of a numeric expression.
CURDIR$	Returns the currently selected directory.
DATE$	Returns a string, containing the system date.
DIM	Declares local variables.
DIR$	Function to read out the entries in a directory.
DO ... LOOP	Loop with condition.
END	Ends the program.
EOF	Checks whether the end of a file was reached.
ERASE	Releases allocated memory of a variable.
ERF$	Returns the name of program file, in which the last error occurred.
ERL	Gives the program line, in which the last error occurred.
ERR	Returns the error code of the most recent error.
ERROR	Creates an error with a specified error number.
EVENTWAIT, EVENTSTATUS	Waits for an event and returns the event code.
FOR ... NEXT	Loop with a specified number of cycles.
FREEFILE	Returns the subsequent free file handle.
FUNCTION	Declares a function.
GLOBAL	Declares global variables.
GOSUB ... RETURN	Branches to the specified label and returns at the next RETURN command.
GOTO	Branches to the specified label.
HEX	Converts a hexadecimal representation of a numeric expression into its decimal equivalent.
HEX$	Transforms a numeric expression into its hexadecimal representation.
IF ... THEN	Checks a condition.
INKEY	Checks whether a softkey was pushed and returns its code.
INPUT	Reads data from an open file or from an open port or waits for a user entry and stores the value entered.
INSTR	Searches for the occurrence of a specific sequence of characters within a string.
KILL	Deletes a file.

LEFT$	Reads a specific part of a string, starting from its beginning.
LEN	Returns the number of characters included in a string.
LET	Assigns an expression to a variable.
LGT	Calculates the common logarithm.
LOCATE	Sets the cursor position on-screen for the subsequent output command.
LOG	Calculates the natural logarithm.
MID$	Reads a specific part of a string.
MKDIR	Creates a new directory.
NAME	Renames an existing file.
OCT	Converts an octal representation of a numeric expression into its decimal equivalent.
OCT$	Transforms a numeric expression into its octal representation.
ON ERROR	Branches to the specified label, if an error occurs during program execution.
OPEN	Opens a file or a port and allows for reading or writing data.
PRINT or OUTPUT	Writes data to an open file or port or displays an expression on-screen.
RANDOMIZE	Initializes the random generator.
REM or `	Comment.
RIGHT$	Reads a specific part of a string, starting from its end.
RMDIR	Removes a directory.
RND	Generates a random number.
SELECT CASE	Selects one of several cases.
SGN	Returns the sign of a numeric expression.
SHELL	Calls the operating system shell in order to perform operating system commands (e.g. so-called pipes).
SIN	Calculates the sine of a numeric expression.
SOFTKEYS	Labels the 4400 softkeys.
SPACE$	Returns a string, filled with a specific number of blanks.
SQRT	Calculates the square root of a numeric expression.

STOP	Debugger breakpoint.
SUB	Declares a procedure (subroutine).
TAN	Returns the tangent of a numeric expression.
TEXTATTR	Sets background color and text size, font and color.
TIME$	Returns a string containing the system time.
VAL	Converts a string expression into a numeric expression.
VAL$	Converts a numeric expression into a string expression.
VARIABLE	Prints variable contents to screen or file.
WAIT	Halts the program for a specified amount of time.

Syntax errors

Code	Message and Comment
1	No program file argument The argument needs to be a valid file name of an existing file.
2	Program file not found The program file was not found in the current or specified directory.
3	Expression expected A numeric expression was expected, but was not found.
4	Variable expected A valid variable name was expected, but was not found.
5	String expected In a context, where only a string is allowed, there was no string found.
6	Number expected In a context, where only a numeric parameter is allowed, there was none found.
7	String const exceeds line The string was not terminated with a quotation mark (").
8	(B)in, (O)ct or (H)ex expected After a '&', only a binary (B), octal (O) or hexadecimal (H) number is allowed.
9	Program line too long Either the program line is more than 254 characters long or the program line continue mark ('_') was used too often.

10	`EOL expected`	The program line should end here. In many cases, the colon (:) that separates two valid commands is missing.
11	`'(' expected`	An open round bracket was expected, but was not found.
12	`')' expected`	A closed round bracket was expected, but was not found.
13	`',' expected`	A comma was expected, but was not found.
14	`'=' expected`	A '=' sign was expected, but was not found.
15	`'#' expected`	A '#' symbol was expected, but was not found. '#'s are used to indicate file handles.
20	`CASE, CASE ELSE or END SELECT without SELECT`	At the beginning of a SELECT block, there must be a SELECT command.
21	`CASE Command:` `Number expected`	When the SELECT block is used to decide on numbers, the CASE statements need to be numeric as well.
22	`CASE Command:` `String expected`	When the SELECT block is used to decide on strings, the CASE statements need to be strings as well.
23	`CASE Command:` `Missing END SELECT`	At the end of a SELECT block, there must be an END SELECT statement.
24	`CASE only at begin of line`	The CASE statements are only allowed at the beginning of a program line.
30	`IF Command:` `Missing END IF`	When using a multiline IF...THEN command, an END IF statement must be used in the last line of the block.
31	`IF Command:` `THEN expected`	Every IF has to be followed by a THEN.
32	`IF Command:` `ELSEIF only at begin of line`	The ELSEIF statements are only allowed at the beginning of a program line.

37	`DO Command:`
	`Missing LOOP`
	A `DO` command needs always to be finished with a `LOOP` statement.

38	`DO Command:`
	`WHILE or UNTIL expected`
	Within a `DO...LOOP` command, only one of the conditions `WHILE` or `UNTIL` may be used – either with `DO` or with `LOOP`.

| 40 | `NEXT Variable and FOR Variable are not the same` |
| | The variable used with the `FOR` statement must be identical with the variable used with the `NEXT` statement of the same block. |

41	`FOR Command:`
	`Missing NEXT`
	A `FOR` command block must be ended with a `NEXT` statement.

42	`FOR Command:`
	`TO expected`
	The `FOR` statement may include a range for the variable to be incremented. This range is specified with `TO` (e.g. `FOR i = 1 TO 10`).

| 46 | `Chain Stack overflow` |
| | The maximum depth of chained programs is 4. In case more than those 4 programs have been chained (e.g. by a chained program that itself chains other programs), this message will occur. |

| 47 | `Chain file not found` |
| | The RAPID! program file that should be chained could not be found in the current or specified directory. |

| 50 | `Expression: NOT,+,- are not allowed here` |
| | Signs may not be used in this context. |

| 51 | `Expression: Command or Function is not allowed here` |
| | A command or a function is not allowed in this context. |

| 52 | `Label expected` |
| | A symbolic label is expected (e.g. after an `ON ERROR`, `GOTO`, `GOSUB` or `RESUME` command). |

| 53 | `Label not found` |
| | The symbolic label issued with a branch command could not be found in the program. Mostly, this is due to typing errors. |

| 54 | `Label already exists` |
| | The symbolic label stated has already been used with a different branch command. |

58	`Variable not declared` In order to access an indexed variable, that one must be declared first, using the `DIM` command.
59	`Variable already declared` This variable was declared already, using a `DIM` command (happens frequently when chaining programs).
60	`Indexvariables are not allowed here` Indexed variables are not allowed in this context (probably a loop can help).

61	`PRINT Command:`
	`',' or ';' expected`
	Several expressions may be handled, using just one `PRINT` command. However, the single expressions need to be separated by commas (,) or semicolons (;).

62	`Keyword not allowed here` The keyword is not allowed in this context. Most of these errors are caused by using a reserved RAPID! name as a symbolic label.
63	`Stack overflow` Too many loops are interlocked. Try to finish one or more of those loops before the subsequent are started.
64, 65	`Misplaced argument` The command or argument is not allowed in this context.
66	`Compare operand expected` The program expects an operand, expressing comparison (=, <, >, <=, >=, <>)

70	`OPEN command:`
	`FOR needs APPEND, BINARI, INPUT, OUTPUT or RANDOM`
	When opening a file with the `FOR` command option, a file mode has to be specified. Currently, only the following three file modes are available: `INPUT`, `OUTPUT` or `APPEND`.

71	`OPEN command:`
	`ACCESS needs READ, WRITE or READ WRITE`
	This message should not occur as the related functionality has not been implemented yet.

72	`OPEN command:`
	`AS expected`
	Indicates a missing file handle.

73	`OPEN command: SHARED, LOCK READ, LOCK WRITE or LOCK READ WRITE expected` This message should not occur as the related functionality has not been implemented yet.
80	`SUB or FUNCTION header error` Please check the declaration of your procedure or function; looks like something is missing there.

81   `SUB or FUNCTION already exist`
There is already a procedure or function with this name.
Many of those cases occur by chaining RAPID! program files.

82   `SUB not declared`
The procedure you want to use currently has not been defined before. In order to avoid that, it is recommended to declare all procedures and functions at the beginning of the main program.

83   `SUB or FUNCTION declaration misplaced`
A procedure or function declaration cannot be used at this place.

84   `SUB END or FUNCTION END does not match to header`
The last program line of any procedure or function must be either an `END SUB` (for procedures) or an `END FUNCTION` (for functions).

85   `SUB EXIT or FUNCTION EXIT does not match to header`
Any procedure must be exited with `EXIT SUB`. Any function must be exited with `EXIT FUNCTION`.

86   `SUB END or FUNCTION END missing`
The last program line of any procedure or function must be either an `END SUB` (for procedures) or an `END FUNCTION` (for functions).

89   `Too many arguments`
More arguments than expected were used.

97   `Internal Error: Parameter`
A procedure or function was called, using the wrong type or the wrong number of arguments.

98   `Internal Error: Command not implemented`
You are trying to use a command that has not been implemented yet. However, the command is a reserved RAPID! name and every attempt to use it will cause this message.

99   `Internal Error: PASS2`
This message indicates an internal error of the program loader. If you receive this message after a restart, please contact a Willtek support center.

Runtime errors

Code   **Message and Comment**

100	Out of memory There is no more memory available for variables. To avoid this situation, please check whether there are indexed variables that are not completely used during runtime or whether one or several indexed variables may be erased using the ERASE command.
101	Too many GOSUBs Overflow of the GOSUB...RETUN stack. To avoid this error message, reduce the number of recursive procedure calls.
102	RETURN without GOSUB A RETURN command was found, but the program is in no procedure currently.
103	VAL: Number expected The VAL function returns a numeric expression (contained in a string expression).
104	Division by zero This is a major problem for program-internal error-handling. To completely avoid this problem, it is recommended to check the operands of a division before the division is carried out.
105, 106	Index out of range The index of an indexed variable must be in the range of -32,766 to +32,766.
107	ON ERROR not allowed in an error handling routine Within program-internal error handling, an ONERROR command is not allowed.
109	Dimension Error The physical dimensions of the operands do not match (e.g. i = 1 kHz + 5 mW).
110	No such file or directory The file name or directory specified does not exist.
111	Too many open files Too many files have been opened and there is no further file handle available. To avoid this message, it is recommended to close all files as soon as they are no loger required to be open.
112	Permission denied There is no permission to access the specified file or directory.
113	Invalid access code The access code used is not valid and access is still denied.

114   `Unexpected File error`
Happens e.g. when you write data to a disk and remove that one during program run. However, it may also indicate problems with the disk drives or controllers. Should this error message occur repeatedly, please contact your nearest support centre.

115   `File not open`
The file specified has not been opened before. Files may only be accessed after they have been opened with the `OPEN` command.

116   `Bad file number`
The file handle used is invalid. This happens for instance, when you try to open a file with the file handle of a file that was opened before.

117   `File already exists`
The file or directory specified does exist and cannot be created anew.

118   `Filesystem is Read Only`
The current file system only allows to read data. This message will occur when you try to write data to a file that is marked 'Read Only'.

119   `A component of Path is not a directory`
The path specified – or parts of it – do not fulfill the naming conventions for directories. Please remember that a directory name must not contain any symbols and is restricted to a maximum length of 8 characters.

120   `Directory busy`
The directory specified is currently in use by another task and cannot be accessed.

121   `Directory not empty`
You tried to remove a directory, but the directory is currently not empty.
Before a directory may be deleted, all entries need to be deleted first.

122   `Filename must not be empty`
The string specifying a file name is empty.

123   `File number already in use`
The current file handle has already been used to open a file and this file is still open. Either close the open file or open the current file with a different file handle. This problem can easily be avoided if you use the `FREEFILE` function to assign file handles.

124   `SUB/FUNCTION Stack overflow`
Overflow of the `GOSUB...RETUN` and `FUNCTION` stacks. To avoid this error message, it is recommended to keep the number of recursive calls of functions and procedures to a minimum.

125	`Open GPIB failed`	The GPIB port could not be opened. Please check whether all parameters used in the `OPEN "GPIB0:..."` command are correct.

125   `Open GPIB failed`
The GPIB port could not be opened. Please check whether all parameters used in the `OPEN "GPIB0:..."` command are correct.

126   `SCPI already open`
The SCPI port may only be opened once.

127   `Input Timeout on COM`
While waiting for an input on one of the serial ports of the 4400, a timeout occurred. This may be due to faulty configuration (either hardware or software) of the handshake signals. Please also check whether all parameters used in the `OPEN "COMx:..."` command are correct.

128   `Output Timeout on COM`
It was not possible to transmit data over the serial ports before the time limit expired. This may be due to faulty configuration (either hardware or software) of the handshake signals. Please also check whether all parameters used in the `OPEN "COMx:..."` command are correct.

199   `USER BREAK`
The program was terminated by the user.

201   `Widget Error: Pos and Size must be defined`
All new widgets need to carry a definition of their position and size. However, widget programming has not been implemented yet.

202   `Widget Error: Widget not found`
The program tried to use a widget that does not exist (anymore). The widget handle is no longer valid. However, widget programming has not been implemented yet.

203   `Widget Error: No valid Widget Call`
The widget call used does not correspond to the widget types available. However, widget programming has not been implemented yet.

SCPI

8

This chapter describes the remote control capabilities of the 4400. Test results 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 4400. 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 4400 via a local area network (TCP/IP Option required).

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 4400 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 4400 than specified in the standard SCPI specification.
However, all SCPI commands implemented on the 4400 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 4400-specific SCPI commands are called Compound commands and will only be understood by the 4400 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 4400, 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 4400).
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 4400 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 4400 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 4400.

– `: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 4400:

– **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 4400 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 4400. Details can be found in subsection "Using queries" on page 349.

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 4400 and sets all system parameters to default values.

A list of all common commands can be found in subsection "Common commands" on page 345.

> **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 4400 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 4400 to factory defaults using SCPI common command `*RST`.
The subsequent command configures the 4400 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 351.

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 4400 is not blocked by other tasks, only perform RAPID! or SCPI measurements from the Welcome menu.

– In order to prevent the 4400 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 4400.

```
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,4400,0511099"`.

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 4400.

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 4400 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 4400 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 4400 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 357.

The use of the service register is outlined in subsection "At the top: the service register" on page 359.

The scheme to use the registers for communication between a RAPID! program or the GPIB on one side and the 4400 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 350 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 4400. These examples concentrate on the following topics:

– "Standard TX measurements" on page 351

– "Message exchange" on page 352

– "Example of a GPIB protocol" on page 353

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 4400	Data	Comment
to	`:PROG:NAME MSG-DEMO.RBS`	Load program
to	`:PROG:STAT RUN`	Start program on 4400
to	`:PROG:MESS "TX, 1, 62, 124, -105"`	Start (user-programmed) TX calibration process on 4400
from	`SRQ`	4400 raises service request after calibration
to	serial poll	Controller identifies SRQ device
from	`SYST:MESS?`	Controller queries system message
to	`"TX Calibration OK"`	4400 confirms successful TX calibration
from	`:PROG:MESS "RX, 1, 62, 124, 15"`	Start (user-programmed) RX calibration process on 4400

to/ from 4400	Data	Comment
	serial poll	Controller identifies SRQ device
to	`SYST:MESS?`	Controller queries system message
from	`"RX Calibration OK"`	4400 confirms successful RX calibration
to	`:MEA-Sure: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`	4400 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 4400

This is a very simplified schematic view of the control and data flow inside the 4400.

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 4400. 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 4400.

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 4400, 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 4400. 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 4400 (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 4400.

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 357 before using the status subsystem.

A table of all registers implemented can be found in subsection "Table of registers" on page 360.

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 4400 starts to page a mobile under test, a certain bit of a certain condition register will be set. When the 4400 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 4400, 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 4400. 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 4400 (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 4400.
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 4400 waits for a trigger.
6	64	Set while the 4400 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 4400. 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 4400.
1	2	Current out of range. Not used on the 4400.
2	4	Time out of range. Not used on the 4400.
3	8	Power out of range. Not used on the 4400.
4	16	Temperature out of range. Not used on the 4400.
5	32	Frequency out of range. Not used on the 4400.
6	64	Phase out of range. Not used on the 4400.
7	128	Modulation out of range. Not used on the 4400.
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 4400.
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 4400'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 4400.
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 4400.
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 4400'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 4400 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 4400 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 4400 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 4400.

> **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 4400 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 4400. 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 4400.

– In case the 4400 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 4400'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 4400 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 4400 does not allow the measurement to be performed or completed), a timeout will occur and an error message will be added to the 4400'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 4400'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 4400 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 365 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 4400's internal error queue (for further details, refer to section ":MEASure[:CONTinuous]" on page 367).
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 4400's internal error queue (for further details, refer to section ":MEASure[:CONTinuous]" on page 367).

Examples	`:MEASure:GSM:ARRay:RFTX:PPEAk 10`
	This command will make the 4400 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 4400 from the mobile)

– phase error

– modulation spectrum of burst received by the 4400 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 355th 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 355th 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 4400 when the Audio Option is installed.

:MEASure:…:GROup   As mentioned before (see **Notes** in section), 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 4400. 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

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 4400 from the mobile)
- phase error
- modulation spectrum of burst received by the 4400 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 4400 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).

```
       ┌─────────────────┐
      (  Start            )        CALCulate:RESet
      (  (CALL ACTIVE)     )
       └────────┬────────┘
                │
       ┌────────┴────────┐
       │ Reset statistics│           CALCulate:LIMit:FAIL:CUMulative:RESet
       │   and limits    │
       └────────┬────────┘
                │
       ┌────────┴────────┐           MEASure:ARRay:RFTX:LENGth? 10
       │   Start two     │
       │  measurements   │           MEASure:ARRay:RFTX:PRMS? 10
       └────────┬────────┘
                │
       ┌────────┴────────┐           CALCulate:MMINimum?
       │   Calculate     │           CALCulate:MMAXimum?
       │ current, average│           CALCulate:MAVerage?
       │  min and max.   │
       └────────┬────────┘           CALCulate:LIMit:FAIL:LAST?
                │
              ╱─┴─╲
            ╱  Last  ╲      No    ┌──────────┐      ┌─────┐
           ╱ measurement╲─────────│  Error   │─────▶(  End  )
           ╲within limits?╱       │ handling │      └─────┘
            ╲    ╱─╲    ╱         └──────────┘
              ╲─┬─╱
                │ Yes                CALCulate:LIMit:FAIL:CUMlative?
              ╱─┴─╲
            ╱   All  ╲      No    ┌──────────┐      ┌─────┐
           ╱ measurement╲─────────│  Error   │─────▶(  End  )
           ╲within limits?╱       │ handling │      └─────┘
            ╲    ╱─╲    ╱         └──────────┘         ▲
              ╲─┬─╱                                    │
                │ Yes                                  │
                └──────────────────────────────────────┘
```

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 4400 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 377.

– 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 4400 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 detailed explanation on those complex limits in subsection "Working with complex limits" on page 381.
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 4400.
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 4400, 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 4400 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 4400 in the base channel can be set or altered using the SCPI commands described in section "The CONFigure subsystem" on page 364.

– A good way to set up the 4400 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 4400 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 4400.

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 4400.

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 4400 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 4400.	
221	Settings of the 4400 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

A

This appendix describes the commands for test automation using RAPID!, the General Purpose Interface Bus (GPIB) 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.

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, 4400, 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	Resets the entire test set. All parameters, limits etc. will be set to the internally pre-defined default values.
Query	There is no query form of this command available.

*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 4400 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 4400.
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 4400 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 Service ReQuest 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 4400 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 4400'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 4400. The queue entry returned will be a string containing the error no. and additional text. The maximum length of the string is 255 characters. **Note:** An overview of all SCPI error messages can be found in section "SCPI command errors".
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 4400. 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 4400. The string returned will contain one integer (and no text). **Note:** An overview of all SCPI error messages can be found in section "SCPI command errors".
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 4400. The string returned will contain a maximum of 10 integers, separated by commas. **Note:** An overview of all SCPI error messages can be found in section "SCPI command errors".
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 4400's internal system **message** queue.
Query	Reads and returns the oldest unread message in the Willtek 4400'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 4400 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 4400. 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 4400.
Description	This command sets the IP address of the 4400. 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 4400.
Description	This command sets the net mask of the 4400. 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 4400.
Description	This command sets the default gateway address of the 4400. 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 4400. The address must be in the range from 49152 to 65535.
Description	This command sets the port address on which the 4400 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 4400 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: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 pre-defined 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|PORTu-`
	`gal|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 4400. 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".

NOTE

For further details on the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357.

: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: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 4400. Consequently, condition-type registers will be updated continuously. **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 4400 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 4400 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. **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 4400. Consequently, condition-type registers will be updated continuously. The bits of the signaling operation event register have the following meaning: Bit 0 (decimal 1): The 4400 is in idle mode Bit 1 (decimal 2): The 4400 is paging the mobile. Bit 2 (decimal 4): A call has been established and is currently active. Bit 3 (decimal 8): The BER loop has been closed. Bit 4 (decimal 16): Async mode (e.g. Generator/Analyzer) active. Bit 5 (decimal 32): The 4400 is calling the mobile under test. A BS call is in progress. Bit 6 (decimal 64): Not used (reserved). Bit 7 (decimal 128): Not used (reserved). Bit 8 (decimal 256): Alerting. Bits 9 through 15: Not used (reserved). **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 BER loop being closed) has been set. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 388.

: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 4400 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.
	Notes
	– For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357.
	– 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 388.

: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 388.

: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 388.

: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 388.

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

Syntax	:STATus:OPERation:SIGNalling:GPRS[:EVENt]?
Parameters	There are no parameters.
Description	Reads out the current contents of the GPRS 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 4400. Consequently, condition-type registers will be updated continuously. **Note:** In case that a certain event shall be trapped in a loop within a program, always query the event-type register. The meaning of the bits is: bit 0 – Set when the mobile is GPRS Attached bit 1 – Set when signaling is in progress bit 2 – Set when the 4400 is ready to measure (PDTCH set up) **Examples** If the status is 0 then the mobile is not GPRS attached. If the status is 1, the mobile is attached (and nothing more). If the status is 3, the mobile is attached and a TBF setup is progressing (but not yet completed). If the status is 5, the mobile is attached and a TBF is set up; measurements can be taken immediately.

Example	:STATus:OPERation:SIGNalling:GPRS:EVENt? Value returned: "5". This means that bits 0 and 2 of the GPRS signaling operation event register have been set: The mobile is GPRS Attached and the PDTCH is set up so the Willtek 4400 is ready to take measurements. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 388.

:STAT:OPERation:SIGNalling:GPRS:CONDition?

Syntax	:STAT:OPERation:SIGNalling:GPRS: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 4400 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:** See :STATus:OPERation:SIGNalling:GPRS[:EVENt] for the meaning of the individual bits of the register.
Example	:STATus:OPERation:SIGNalling:GPRS:CONDition? Value returned: "3". This means that bits 0 and 1 have been set. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 388.

:STATus:OPERation:SIGNalling:GPRS:ENABLe

Syntax	:STATus:OPERation:SIGNalling:GPRS: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 GPRS Signaling Operation Register Group. This mask will be **AND**ed with the GPRS signaling operation **event** register and thus decide what kind of events will be forwarded to bit 10 of the General Operation Register Group. See :STATus:OPERation:SIGNalling:GPRS[:EVENt] for the meaning of the individual bits of the register.
Query	There is no query form of this command available.
Example	:STATus:OPERation:SIGNalling:GPRS:ENABLe 4 This means that any setting of bit 2 (PDTCH set up, ready for measurements) of the GPRS signaling operation event register will rise bit 10 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 388.

:STAT:OPERation:SIGNalling:GPRS:PTRansition

Syntax	`:STAT:OPERation:SIGNalling:GPRS: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 GPRS Signaling Operation Register Group. This mask will be **AND**ed with the up to 15 bits of the PTR filter of the GPRS 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:GPRS: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 388.

:STAT:OPERation:SIGNalling:GPRS:NTRansition

Syntax	`:STAT:OPERation:SIGNalling:GPRS: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 GPRS Signaling Operation Register Group. This mask will be **AND**ed with the up to 15 bits of the NTR filter of the GPRS 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:GPRS: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 388.

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

Syntax	`:STATus:OPERation:SIGNalling:EGPRs[:EVENt]?`
Parameters	There are no parameters.
Description	Reads out the current contents of the GPRS 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 4400. 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.
	The meaning of the bits is:
	bit 0 - Set when the mobile is GPRS Attached
	bit 1 - Set when signaling is in progress
	bit 2 - Set when the 4400 is ready to measure (PDTCH set up)
Example	`:STATus:OPERation:SIGNalling:EGPRs:EVENt?`
	Value returned in this example: "5".
	This means that bits 0 and 2 of the GPRS signaling operation event register have been set: The mobile is GPRS Attached and the PDTCH is set up so the Willtek 4400 is ready to take measurements. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 388.

`:STAT:OPERation:SIGNalling:EGPRs:CONDition?`

Syntax	`:STAT:OPERation:SIGNalling:EGPRs:CONDition?`
Parameters	There are no parameters.
Description	This command reads out the current contents of the GPRS Signaling Operation Condition Register. This register reflects the current state of the Willtek 4400 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:
	See `:STATus:OPERation:SIGNalling:EGPRs[:EVENt]` for the meaning of the individual bits of the register.
Example	`:STATus:OPERation:SIGNalling:EGPRs:CONDition?`
	Value returned in this example: "3".
	This means that bits 0 and 1 have been set. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 388.

`:STATus:OPERation:SIGNalling:EGPRs:ENABLe`

Syntax	`:STATus:OPERation:SIGNalling:EGPRs: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 GPRS Signaling Operation Register Group. This mask will be **AND**ed with the GPRS signaling operation **event** register and thus decide what kind of events will be forwarded to bit 10 of the General Operation Register Group.
	See `:STATus:OPERation:SIGNalling:EGPRs[:EVENt]` for the meaning of the individual bits of the register.
Query	There is no query form of this command available.
Example	`STATus:OPERation:SIGNalling:EGPRs:ENABle 4`
	This means that any setting of bit 2 (PDTCH set up, ready for measurements) of the GPRS signaling operation event register will raise bit 10 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 388.

`:STAT:OPERation:SIGNalling:EGPRs:PTRansition`

Syntax	`:STAT:OPERation:SIGNalling:EGPRs: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 GPRS Signaling Operation Register Group. This mask will be **AND**ed with the up to 15 bits of the PTR filter of the GPRS 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:EGPRs: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 388.

`:STAT:OPERation:SIGNalling:EGPRs:NTRansition`

Syntax	`:STAT:OPERation:SIGNalling:EGPRs: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 GPRS Signaling Operation Register Group. This mask will be **AND**ed with the up to 15 bits of the NTR filter of the GPRS 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:EGPRs: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 388.

: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 4400. Consequently, the condition type registers will be updated continuously. **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section Understanding the STATus Subsystem. – 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 4400 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. **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 4400. Consequently, condition-type registers will be updated continuously. **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 4400 is out of range. If you want to know whether the Willtek 4400 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 4400 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** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 4400.

: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 4400. Consequently, condition-type registers will be updated continuously. **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 4400 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. **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 4400 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 4400. Consequently, condition-type registers will be updated continuously. **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 4400.

: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 4400 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. **Notes** – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 357. – 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 4400.

`: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. For more details regarding RAPID!, please refer to chapter "RAPID!" on page 261.
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. For more details regarding floating point real numbers, please refer to section "Variables" on page 282.
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. For more details regarding strings, please refer to section "Variables" on page 282.
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:\4400\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. For details regarding RAPID!, please refer to chapter "RAPID!" on page 261.
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. For more details regarding the status subsystem of the Willtek 4400, please refer to section "Understanding the STATus subsystem" on page 357.
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. For more details regarding the status subsystem of the Willtek 4400, please refer to section "Understanding the STATus subsystem" on page 357.
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 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: ...	These are the commands to set specific system parameters like the base station's RF output power level or its identity.
CONFigure:<SystemOption> :MSTAtion:...	The mobile-specific information is handed over to the 4400 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 4400.

NOTE

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

`: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|AMPS|TDSCdma.` `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 4400.
	GENana will make the Willtek 4400 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 4400 will generate and analyze circuit-switched GSM signals only with this setting.
	GCGenana will set up the Willtek 4400 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 4400 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 4400 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 4400 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: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"**

<div style="text-align: center">

:CONFigure:COUPloss:NAME

</div>

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"**

<div style="text-align: center">

:CONFigure:COUPloss:INFormation?

</div>

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"**

<div style="text-align: center">

:CONFigure:COUPloss:DATA

</div>

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 4400.
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 4400 frequency reference is synchronized to an external 10 MHz clock signal fed into the EXT SYNC connector on the rear panel.

:CONFigure:GPRS:ANALyzer:LEVel:EXPect[:DATA]

Syntax	:CONFigure:GPRS:ANALyzer:LEVel:EXPect[:DATA] <int1>
Parameters	int1 is an integer. The minimum value for int1 is −10, the maximum value is 30. The default value for int1 is 10.
Description	Sets the RF power level at which the mobile under test is going to transmit. The physical dimension of int1 is dBm. **Note:** Specifying an expected power level in analyzer mode speeds up testing as it avoids or shortens auto ranging and auto triggering procedures.

Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:ANALyzer:LEVel:EXPect[:DATA] 20`
	`:CONF:GPRS:ANAL:LEV:EXP?`
	Value returned: `"20"`. This means that the Willtek 4400 will expect the mobile to send its data at an RF power level of 20 dBm.

:CONFigure:GPRS:ANALyzer:LEVel:EXPect:MODE

Syntax	`:CONFigure:GPRS:ANALyzer:LEVel:EXPect:MODE <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON│OFF`. Default is `OFF`.
Description	Switches the use of the 'expected level feature' on or off. Please refer to the description of the `:CONF:GPRS:ANAL:LEV:EXP` command above for details.
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	`:CONFigure:GPRS:ANALyzer:LEVel:EXPect:MODE ON`
	`:CONF:GPRS:ANAL:Lev:EXP:MODE?`
	Value returned: `"ON"`.

:CONFigure:GPRS:BS:RCL

Syntax	`:CONFigure:GPRS:BS:RCL <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is 1, the maximum value is 7. The default value for `int1` is 1.
Description	This command sets the routing area color code. This code is used to indicate to the mobile that adjacent cells belong to different routing areas. This parameter is broadcast by the Willtek 4400 in the SI3 and SI4 rest octets.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:RCL 3`
	`:CONF:GPRS:BS:RCL?`
	Value returned: `"3"`

:CONFigure:GPRS:BS:RCO

Syntax	`:CONFigure:GPRS:BS:RCO <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 routing area code. This code is used to indicate the routing area, the base station belongs to. This parameter is broadcast by the Willtek 4400 in the SI13 rest octets.

Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:RC0 124`
	`:CONF:GPRS:BS:RCO?`
	Value returned: `"124"`

:CONFigure:GPRS:BS:ALPHa

Syntax	`:CONFigure:GPRS:BS:ALPHa <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is 0, the maximum value is 15. The default value for `int1` is 0.
Description	This command sets the alpha factor in (asynchronous) generator/analyzer mode (please refer to the description of the `:CONF:CSYS:GPG` command for further information). The alpha factor is one of the GPRS power control parameters. It is required by the mobile to calculate the uplink power level it has to use on a PDCH (packed data channel). The alpha factor is broadcast by the Willtek 4400 in the SI13 rest octets.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:ALPHa 10`
	`:CONF:GPRS:BS:ALPH?`
	Value returned: `"10"`

:CONFigure:GPRS:BS:RLCMac:USF

Syntax	`:CONFigure:GPRS:BS:RLCMac:USF <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 uplink state flag (USF) in (asynchronous) generator/analyzer mode (please refer to the description of the `:CONF:CSYS:GPG` command for further information). The USF is used to either assign a block to a particular mobile (one out of seven possible) or to mark it as unassigned. All time slots will use the same USF. The USF is part of the MAC header on the PDTCH.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:RLCMac:USF 0`
	`:CONF:GPRS:BS:RLCM:USF?`
	Value returned: `"0"`

:CONFigure:GPRS:BS:RLCMac:USF:INC

Syntax	`:CONFigure:GPRS:BS:RLCMac:USF:INC <PredefExp>`

Parameters	`PredefExp` is one of the following predefined expressions: `ON`|`OFF`. Default is `ON`.
Description	Switches the auto-incrementing of the USF either on or off. For further details regarding the USF, please refer to the description of the `:CONFigure:GPRS:BS:RLCMac:USF` command above.
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	`:CONFigure:GPRS:BS:RLCMac:USF:INC OFF`
	`:CONF:GPRS:BS:RLCM:USF:INC?`
	Value returned: `"OFF"`.

:CONFigure:GPRS:BS:RLCMac:RRBP[:DATA]

Syntax	`:CONFigure:GPRS:BS:RLCMac:RRBP[:DATA] <PredefExp1>`
Parameters	`PredefExp1` is one of the following predefined expressions: `N13`|`N17`|`N21`|`26`. Default is `N13`.
Description	This command sets the relative reserved block period (RRBP) in (asynchronous) generator/analyzer mode (please refer to the description of the `:CONF:CSYS:GPG` command for further information). The RRBP marks the block to which the USF parameter corresponds, in relation to the block currently used.
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	`:CONFigure:GPRS:BS:RLCMac:RRBP:DATA N21`
	`:CONF:GPRS:BS:RLCM:RRBP?`
	Value returned: `"N21"`

:CONFigure:GPRS:BS:RLCMac:RRBP:MODE

Syntax	`:CONFigure:GPRS:BS:RLCMac:RRBP:MODE <PredefExp1>`
Parameters	`PredefExp1` is one of the following predefined expressions: `ON`|`OFF`. Default is `ON`.
Description	This command indicates whether the value specified for the relative reserved block period (RRBP) shall be used by the mobile (`ON`) or not (`OFF`) in (asynchronous) generator/analyzer mode (please refer to the description of the `:CONF:CSYS:GPG` command for further information).
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	`:CONFigure:GPRS:BS:RLCMac:RRBP:MODE OFF`
	`:CONF:GPRS:BS:RLCM:RRBP:MODE?`
	Value returned: `"OFF"`

:CONFigure:GPRS:BS:RLCMac:PR

Syntax	:CONFigure:GPRS:BS:RLCMac:PR <PredefExp1>
Parameters	PredefExp1 is one of the following predefined expressions: RNG6|RNG14|RNG22|RNG30. Default is RNG6.
Description	This command specifies the power reduction (PR) in (asynchronous) generator/analyzer mode (please refer to the description of the :CONF:CSYS:GPG command for further information). The power reduction the reduction in RF power (in dB) from the BCCH to the PDCH in downlink. PredefExp1 indicates the range of the power reduction: RNG6 stands for a power reduction of 0...6 dB, RNG 14 indicates a power reduction of 8...14 dB, RNG22 stands for a power reduction of 16...22 dB, while RNG30 indicates a power reduction of 24...30 dB.
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	:CONFigure:GPRS:BS:RLCMac:RRBP:PR RNG6 :CONF:GPRS:BS:RLCM:RRBP:PR? Value returned: "RNG6"

:CONFigure:GPRS:BS:RLCMac:TFI

Syntax	:CONFigure:GPRS:BS:RLCMac:TFI <int1>
Parameters	int1 is an integer. The minimum value for int1 is 0, the maximum value is 31. The default value for int1 is 0.
Description	This command sets the temporary flow identifier (TFI) in (asynchronous) generator/analyzer mode (please refer to the description of the :CONF:CSYS:GPG command for further information).
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:GPRS:BS:RLCMac:TFI 15 :CONF:GPRS:BS:RLCM:TFI? Value returned: "15"

:CONFigure:GPRS:BS:RLCMac:BSN[:DATA]

Syntax	:CONFigure:GPRS:BS:RLCMac:BSN[:DATA] <int1>
Parameters	int1 is an integer. The minimum value for int1 is 0, the maximum value is 127. The default value for int1 is 0.
Description	This command sets the block sequence number (BSN) in (asynchronous) generator/analyzer mode (please refer to the description of the :CONF:CSYS:GPG command for further information). The BSN is the absolute block sequence number modulo 128 of each RLC data block within the TBF. It is used to enumerate the RLC data blocks, starting with 1.

Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:RLCMac:BSN:DATA 1`
	`:CONF:GPRS:BS:RLCM:BSN?`
	Value returned: `"1"`

`:CONFigure:GPRS:BS:RLCMac:BSN:INC`

Syntax	`:CONFigure:GPRS:BS:RLCMac:BSN:INC <PredefExp1>`
Parameters	`PredefExp1` is one of the following predefined expressions: `ON`|`OFF`. Default is `ON`.
Description	This command switches the auto-increment of the block sequence number either on (`ON`) or off (`OFF`) as long as the Willtek 4400 is working in (asynchronous) generator/analyzer mode (please refer to the description of the `:CONF:CSYS:GPG` command for further information).
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	`:CONFigure:GPRS:BS:RLCMac:BSN:INC OFF`
	`:CONF:GPRS:BS:RLCM:BSN:INC?`
	Value returned: `"OFF"`

`:CONFigure:GPRS:BS:ATTach`

Syntax	`:CONFigure:GPRS: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 4400 when switched on or off, respectively. This can be used to detect when the mobile is switched on and listening to the Willtek 4400 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:GPRS:BS:ATTach OFF`
	`:CONF:GPRS:BS:ATT?`
	Value returned: `"OFF"`.

`:CONFigure:GPRS:BS:MSTAtion:MSLot`

Syntax	`:CONFigure:GPRS:BS:MSTAtion:MSLot <int1>`
Parameters	`int1` is an integer parameter. The minimum value for `int1` is 1, the maximum value is 4. The default value for `int1` is 1.

Description	This command either sets the number of UL time slots expected by the Willtek 4400 (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 4400.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:MSTAtion:MSLot 4`
	 `:CONF:GPRS:MSTA:MSL?`
	 Value returned: `"4"`.

:CONFigure:GPRS:BS:BCC

Syntax	`:CONFigure:GPRS: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:BCC 1`
	 `:CONF:GPRS:BS:BCC?`
	 Value returned: `"1"`.

:CONFigure:GPRS:BS:BCH:ARFCn

Syntax	`:CONFigure:GPRS: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 4400's base channel (BCCH). **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:GPRS:BS:BCH:ARFCn 60`
	 `:CONF:GPRS:BS:BCH:ARFC?`
	 Value returned: `"60"`.

:CONFigure:GPRS:BS:CBA

Syntax	`:CONFigure:GPRS: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:GPRS:BS:CBA 1`
	`:CONF:GPRS:BS:CBA?`
	Value returned: `"1"`.

:CONFigure:GPRS:BS:CI

Syntax	`:CONFigure:GPRS: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:CI 127`
	`:CONF:GPRS:BS:CI?`
	Value returned: `"127"`.

:CONFigure:GPRS:BS:LAI:MCC

Syntax	`:CONFigure:GPRS: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:LAI:MCC 238`
	`:CONF:GPRS:BS:LAI:MCC?`
	Value returned: `"238"`.

:CONFigure:GPRS:BS:LAI:MNC

Syntax	`:CONFigure:GPRS:BS:LAI:MNC <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.

Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:LAI:MNC 20`
	`:CONF:GPRS:BS:LAI:MNC?`
	Value returned: `"20"`.

:CONFigure:GPRS:BS:LAI:MNC:FORMat

Syntax	`:CONFigure:GPRS:BS:LAI:MNC:FORMat <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `TWODig-its|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:GPRS:BS:LAI:MNC:FORMat THRE`
	`:CONF:GPRS:BS:LAI:MNC:FORM?`
	Value returned: `"THRE"`.

:CONFigure:GPRS:BS:LAI:LAC

Syntax	`:CONFigure:GPRS: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:LAI:LAC 17399`
	`:CONF:GPRS:BS:LAI:LAC?`
	Value returned: `"17399"`.

:CONFigure:GPRS:BS:LEVel

Syntax	`:CONFigure:GPRS:BS: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 resolution of `real1` is `0.1`. The default value for `real1` is `-60.0`.
Description	Sets the (standard) output level of the Willtek 4400. If individual levels are selected for GPRS tests, this level is used on the base channel. Applies to GSM/GPRS synchronous mode.
Query	The query form of this command will return the current setting. The string delivered back will contain one real number.

Example	`:CONFigure:GPRS:BS:LEVel -100.0`
	`:CONF:GPRS:BS:LEVel?`
	Value returned: `"-100.0"`.

:CONFigure:GPRS:BS:MSLot[:DATA]

Syntax	`:CONFigure:GPRS:BS:MSLot[:DATA] <int1>`
Parameters	`int1` is an integer parameter. The minimum value for `int1` is 1, the maximum value is 2. The default value for `int1` is 1.
Description	Sets the number of PDTCH channels (slots) to be sent in the downlink. Applies to GPRS synchronous mode.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:MSLot[:DATA] 1`
	`:CONF:GPRS:BS:MSLot[:DATA]?`
	Value returned: `"1"`.

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

Syntax	`:CONFigure:GPRS:BS:MSLot:LEVel[:DATA] <real1>, <real2>,` `<real3>, <real4>`
Parameters	`real1` to `real4` are 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`.
Description	Sets individual downlink slot levels for active (PDTCH) slots in Multislot GPRS operation (BCH remains at the level set by `CONFigure:GPRS:BS:LEVel`. Applies to GPRS synchronous test mode. Data are used only if `CONFigure:GPRS:BS:MSLot:LEVel:MODE` is set to `INDL`.
Query	The query form of this command will return the current setting. The string delivered back will contain four real numbers.

Example	`:CONFigure:GPRS:BS:MSLot:LEVel:DATA -80.0,-20.5,-60.0,`
	`-60.0`
	`:CONF:GPRS:BS:MSL:LEV:DATA?`
	Values returned: `"-80.0,-20.5,-60.0,-60.0"`.

:CONFigure:GPRS:BS:MSLot:LEVel:MODE

Syntax	`:CONFigure:GPRS:BS:MSLot:LEVel:MODE <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `INDLev`|`STDLev`. Default is `INDLev`.

Description	Selects between individual slot levels and standard slot level for PDTCHs. If `STDLev` is selected, the Willtek 4400 transmits at the level selected by `CONFigure:GPRS:BS:LEVel`. If `INDLev` is selected, the Willtek 4400 uses the power levels selected with `CONFigure:GPRS:BS:MSLot:LEVel` for the up to four used time slots; time slot 0 is transmitted at the level selected by `CONFigure:GPRS:BS:LEVel`, and the level selected by `CONFigure:GPRS:BS:MSLot:UNUSed` for the other slots.
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:GPRS:BS:MSLot:LEVel:MODE INDLev`
	`:CONF:GPRS:BS:MSL:LEV:MODE?`
	Value returned: `"INDLev"`.

:CONFigure:GPRS:BS:MSLot:LEVel:UNUSed

Syntax	`:CONFigure:GPRS:BS:MSLot:LEVel:UNUSed <real1>`
Parameters	`real1` is a floating point real number. The minimum value for `real1` is `-120.0`, the maximum value is `-10.0`. The resolution of `real1` is `0.1`. The default value for `real1` is `-60.0`.
Description	Sets the power level that the Willtek 4400 transmits in the unused time slots in packet data mode (i.e. except in time slot 0 and the assigned PDCH slots). Applies to GPRS synchronous mode.
Query	The query form of this command will return the current setting. The string delivered back will contain one real number.

Example	`:CONFigure:GPRS:BS:MSLot:LEVel:UNUSed -60.0`
	`:CONF:GPRS:BS:MSL:LEV:UNUS?`
	Value returned: `"-60.0"`.

:CONFigure:GPRS:BS:NCC

Syntax	`:CONFigure:GPRS: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:NCC 2`
	`:CONF:GPRS:BS:NCC?`
	Value returned: `"2"`.

:CONFigure:GPRS:BS:NCEL

Syntax	`:CONFigure:GPRS:BS:NCEL1` `[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 4400. The mobile under test has to perform a MS report on those frequency channels. **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 4400 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 4400 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:GPRS:BS:NCEL1 10,20,30,40,50,60`
	`:CONF:GPRS:BS:NCEL 70,80`
	`:CONF:GPRS:BS:NCEL?`
	Values returned: `"70,80,30,40,50,60"`.

:CONFigure:GPRS:BS:ORIGinate

Syntax	`:CONFigure:GPRS:BS:ORIGinate <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ONRequest`|`ONLOcation`. Default is `ONRequest`.
Description	In GSM/GPRS, a TCH can be assigned either on request or as soon as the mobile has been registered (= immediately after a location update). `ONRequest` will assign a TCH only when either the mobile under test or the Willtek 4400 initializes a call. `ONLOcation` will assign a TCH as soon as the registration has been completed.
Query	The query form of this command will return the current setting. The string delivered back will contain the short-form version one of the predefined expressions as explained above.

Example	`:CONFigure:GPRS:BS:ORIGinate ONLOcation`
	`:CONF:GPRS:BS:ORIG?`
	Value returned: `"ONLO"`.

:CONFigure:GPRS:BS:PDTCh:ARFCn

Syntax	`:CONFigure:GPRS:BS:PDTCh: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 `2`.
Description	This command sets the frequency channel number of the simulated packet data traffic channel.

Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:PDTCh 2`
	`:CONF:GPRS:BS:PDTC?`
	Value returned: `"2"`.

`:CONFigure:GPRS:BS:TCH:ARFCn`

Syntax	`:CONFigure:GPRS: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:TCH:ARFCn 917`
	`:CONF:GPRS:BS:TCH:ARFC?`
	Value returned: `"917"`.

`:CONFigure:GPRS:BS:TCH:TSLot`

Syntax	`:CONFigure:GPRS: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) or the PDTCH.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:BS:TCH:TSLot 5`
	`:CONF:GPRS:BS:TCH:TSL?`
	Value returned: `"5"`.

`:CONFigure:GPRS:BS:TCH:TYPE`

Syntax	`:CONFigure:GPRS: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.
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:GPRS:BS:TCH:TYPE EFR :CONF:GPRS:BS:TCH:TYPE? Value returned: "EFR".

:CONFigure:GPRS:BLER:COUNt

Syntax	:CONFigure:GPRS:BLER:COUNt <int1>
Parameters	int1 is an integer parameter. The minimum value for int1 is 0, the maximum value is 10000. The default value for int1 is 1.
Description	Defines the number of samples (blocks) for each BLER measurement. Applies to BLER measurements in GPRS synchronous mode.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:GPRS:BLER:COUNt 1 :CONF:GPRS:BLER:COUNt? Value returned: "1".

:CONFigure:GPRS:BER:LOOP

Syntax	:CONFigure:GPRS: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 4400. 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 4400. 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 4400.
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:GPRS:BER:LOOP RESidual :CONF:GPRS:BER:LOOP? Value returned: "RES".

:CONFigure:GPRS:BER:BITPattern

Syntax	:CONFigure:GPRS: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:GPRS:BER:BITPattern ALLZero` `:CONF:GPRS:BER:BITP?` Value returned: "ALLZ".

:CONFigure:GPRS:BER:COUNt

Syntax	`:CONFigure:GPRS:BER:COUNt <long1>`
Parameters	long1 is a long integer number. The minimum value for long1 is 2,000, the maximum value is 100,000. The default value for long1 is 1,000,000.
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:GPRS:BER:COUNt 100000` `:CONF:GPRS:BER:COUN?` Value returned: "100000".

:CONFigure:GPRS:BER:RTDelay

Syntax	`:CONFigure:GPRS: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:GPRS:BER:RTDelay 9` `:CONF:GPRS:BER:RTD?` Value returned: "9"

:CONFigure:GPRS:MSTAtion:DRX

Syntax	`:CONFigure:GPRS: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). **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:GPRS:MSTAtion:DRX 2` `:CONF:GPRS:MSTA:DRX?` Value returned: `"2"`, resulting in a DRX period of four multiframes.

:CONFigure:GPRS:MSTAtion:TADVance

Syntax	`:CONFigure:GPRS: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	`:CONFigure:GPRS:MSTAtion:TADVance 12` `:CONF:GPRS:MSTA:TADV?` Value returned: `"12"`.

:CONFigure:GPRS:MSTAtion:PLEVel[:ALL]

Syntax	`:CONFigure:GPRS: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:GPRS:MSTAtion:PLEVel:ALL 17 :CONF:GPRS:MSTA:PLEV? Value returned: "17".

:CONFigure:GPRS:MSTAtion:MODE

Syntax	:CONFigure:GPRS: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.
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:GPRS:BS:CMODe FACCh :CONF:GPRS:BS:CMOD? Value returned: "FACC".

:CONFigure:GPRS:MSTAtion:MSLot

Syntax	:CONFigure:GPRS: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 sets the number of UL time slots that the mobile under test is commanded to use during multislot testing. **Note:** HSCSD testing requires the Multislot Option to be installed on the Willtek 4400.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:GPRS:MSTAtion:MSLot 4 :CONF:GPRS:MSTA:MSL? Value returned: "4".

:CONFigure:GPRS:MSTAtion:LOOPback

Syntax	:CONFigure:GPRS:MSTAtion:LOOPback <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: ON|OFF. Default is ON.
Description	Selects the loopback mode. If the parameter PredefExp is set to ON, the mobile station will support GPRS loopback (mode B), if the parameter PredefExp is set to ON, the mobile station will not support GPRS loopback (mode A).
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:GPRS:MSTAtion:LOOPback ON
:CONF:GPRS:MSTA:LOOP?
Value returned: "ON". |

:CONFigure:GPRS:MSTAtion:PSLots

Syntax	:CONFigure:GPRS:MSTAtion:PSLots <int1>, <int2>, <int3>, <int4>
Parameters	int1 to int4 are integer parameters. The minimum value for each intx is –26, the maximum value is 39. The resolution of each intx is 2, the default value for each intx is 5.
Description	Sets the uplink power level in the range of -23 ... +39 dBm (GSM900) or -26 ... +36 dBm (upper band). Applies to GPRS synchronous mode.
Query	The query form of this command will return the current setting. The string delivered back will contain four integers.
Example	:CONFigure:GPRS:MSTAtion:PSLots –10,–6,0,6 :CONF:GPRS:MSTA:PSL? Value returned: "–10,–6,0,6".

:CONFigure:GPRS:TYPE

Syntax	:CONFigure:GPRS: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 4400 was brought into GSM call mode. GSM9001800 will make the Willtek 4400 work as a test set for the GSM system types GSM 900 and GSM 1800. GSM9001900 will set up the Willtek 4400 as a test set for the GSM system types GSM 900 and GSM 1900. **Notes:** This command will have no effect when the Willtek 4400 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 regions.
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:GPRS:TYPE GSM9001900 :CONF:GPRS:TYPE? Value returned: "GSM9001900".

:CONFigure:GSM:SMS:MCLass

| Syntax | :CONFigure:GSM:SMS:MCLass <int1> |

Parameters	`int1` is an integer. The minimum value for `int1` is 0, the maximum value is 3. The default value for `int1` is 0.
Description	This command allows to predefine the SMS data coding scheme message class.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GSM:SMS:MCLass 1`
	`:CONF:GSM:SMS:MCL?`
	Value returned in this example: `"1"`#

`:CONFigure:GPRS:MEASure:GROUp[:RFTX]`

Syntax	`:CONFigure:GPRS: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 you to set a user-specific sequence of RF TX tests on the Willtek 4400. The sequence has to be started with the appropriate commands of the `MEASure` subsystem (`:MEAS:GPRS:CONT:RFTX:GROU` or `:MEAS:GPRS: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. `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 4400. The string delivered back will contain a maximum of nine short-form versions of the predefined expressions explained above.

Example	`:CONFigure:GPRS:MEASure:GROUp:RFTX PRMS,POWer,FLATness`
	`:CONF:GPRS:MEAS:GROU?`
	Value returned: `"PRMS,POW,FLAT"`

:CONFigure:GPRS:MEASure:GROUp:AFANalyser

Syntax	:CONFigure:GPRS: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 you to set a user-specific sequence of AF tests on the Willtek 4400. 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 the Willtek 4400.
Query	The query form of this command will return the current sequence of AF tests to be performed by the Willtek 4400. The string delivered back will contain a maximum of six predefined short-form versions of the expressions explained above.
Example	:CONFigure:GPRS:MEASure:GROUp:AFANalyser SINad, DISTortion, FREQuency :CONF:GPRS:MEAS:GROU:AFAN? Value returned: "SIN,DIST,FREQ"

:CONFigure:GPRS:MEASure:ACPM:TRANsient

Syntax	:CONFigure:GPRS: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 4400 calculates the RF output spectrum due to switching transients (or ACP Transient for short). If FULL is selected, the Willtek 4400 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:GPRS:MEASure:ACPM:TRAN FULL`
	`:CONF:GPRS:MEASure:ACPM:TRAN?`
	In this example, the query will return the string `FULL`.

:CONFigure:GPRS:MSLot:RESult

Syntax	`:CONFigure:GPRS:MSLot:RESult <int1>`
Parameters	`int1` is an integer parameter. The minimum value for `int1` is 1, the maximum value is 4. The default value for `int1` is 1.
Description	Sets the number of the burst for which to deliver measurement results. Delivers the results for the selected slot out of the number of all measured slots. This command can be used in GPRS synchronous mode when the mobile transmits on multiple time slots but results are fetched from one (selectable) time slot only.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:GPRS:MSLot:RESult 1`
	`:CONF:GPRS:MSLot:RESult?`
	Value returned: `"1"`.

:CONFigure:GPRS:IDRequest

Syntax	`:CONFigure:GPRS:IDRequest <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `OFF`|`ON`. Default is `OFF`.
Description	Switches the ID Request procedure during the Attach on or off
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:GPRS:IDRequest ON`
	`:CONF:GPRS:IDRE?`
	Value returned in this example: `"ON"`.

:CONFigure:EGPRs:ANALyzer:LEVel:EXPect[:DATA]

Syntax	`:CONFigure:EGPRs:ANALyzer:LEVel:EXPect[:DATA] <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is −10, the maximum value is 30. The default value for `int1` is 10.
Description	Sets the RF power level at which the mobile under test is going to transmit. The physical dimension of `int1` is dBm. **Note:** Specifying an expected power level in analyzer mode speeds up testing as it avoids or shortens auto ranging and auto triggering procedures.

Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:EGPRs:ANALyzer:LEVel:EXPect[:DATA] 20`
	`:CONF:EGPRs:ANAL:LEV:EXP?`
	Value returned in this example: "20". This means that the Willtek 4400 will expect the mobile to send its data at a RF power level of 20 dBm.

:CONFigure:EGPRs:ANALyzer:LEVel:EXPect:MODE

Syntax	`:CONFigure:EGPRs:ANALyzer:LEVel:EXPect:MODE <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON`|`OFF`. Default is `OFF`.
Description	Switches the use of the 'expected level feature' ON or OFF. Please refer to the description of the `:CONF:EGPRs:ANAL:LEV:EXP` command above for details.
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	`:CONFigure:EGPRs:ANALyzer:LEVel:EXPect:MODE ON`
	`:CONF:EGPRs:ANAL:Lev:EXP:MODE?`
	Value returned in this example: "ON".

:CONFigure:EGPRs:BS:RCL

Syntax	`:CONFigure:EGPRs:BS:RCL <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is 1, the maximum value is 7. The default value for `int1` is 1.
Description	This command sets the routing area color code. This code is used to indicate to the mobile that adjacent cells belong to different routing areas. This parameter is broadcast by the Willtek 4400 in the SI3 and SI4 rest octets.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:EGPRs:BS:RCL 3`
	`:CONF:EGPRs:BS:RCL?`
	Value returned in this example: "3"

:CONFigure:EGPRs:BS:RCO

Syntax	`:CONFigure:EGPRs:BS:RCO <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 routing area code. This code is used to indicate the routing area, the base station belongs to. This parameter is broadcast by the Willtek 4400 in the SI13 rest octets.

Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`CONFigure:EGPRs:BS:RC0 124`
	`:CONF:EGPRs:BS:RCO?`
	Value returned in this example: `"124"`

:CONFigure:EGPRs:BS:ALPHa

Syntax	`:CONFigure:EGPRs:BS:ALPHa <int1>`
Parameters	`int1` is an integer. The minimum value for `int1` is 0, the maximum value is 15. The default value for `int1` is 0.
Description	This command sets the alpha factor in (asynchronous) generator/analyzer mode (please refer to the description of the `:CONF:CSYS:GPG` command for further information). The alpha factor is one of the GPRS power control parameters. It is required by the mobile to calculate the uplink power level it has to use on a PDCH (packed data channel). The alpha factor is broadcast by the Willtek 4400 in the SI13 rest octets.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:EGPRs:BS:ALPHa 10`
	`:CONF:EGPRs:BS:ALPH?`
	Value returned in this example: `"10"`

:CONFigure:EGPRs:BS:RLCMac:USF

Syntax	`:CONFigure:EGPRs:BS:RLCMac:USF <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 uplink state flag (USF) in (asynchronous) generator/analyzer mode (please refer to the description of the `:CONF:CSYS:GPG` command for further information). The USF is used to either assign a block to a particular mobile (one out of seven possible) or to mark it as unassigned. All time slots will use the same USF. The USF is part of the MAC header on the PDTCH
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:EGPRs:BS:RLCMac:USF 0`
	`:CONF:EGPRs:BS:RLCM:USF?`
	Value returned in this example: `"0"`

:CONFigure:EGPRs:BS:RLCMac:USF:INC

Syntax	:CONFigure:EGPRs:BS:RLCMac:USF:INC <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: ON│OFF. Default is ON.
Description	Switches the auto-incrementing of the USF either ON or OFF. For further details regarding the USF, please refer to the description of the :CONFigure:EGPRs:BS:RLCMac:USF command above.
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	:CONFigure:EGPRs:BS:RLCMac:USF:INC OFF :CONF:EGPRs:BS:RLCM:USF:INC? Value returned in this example: "OFF".

:CONFigure:EGPRs:BS:RLCMac:RRBP[:DATA]

Syntax	:CONFigure:EGPRs:BS:RLCMac:RRBP[:DATA] <PredefExp1>
Parameters	PredefExp1 is one of the following predefined expressions: N13│N17│N21│26. Default is N13.
Description	This command sets the relative reserved block period (RRBP) in (asynchronous) generator/analyzer mode (please refer to the description of the :CONF:CSYS:GPG command for further information). The RRBP marks the block to which the USF parameter corresponds, in relation to the block currently used.
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	:CONFigure:EGPRs:BS:RLCMac:RRBP:DATA N21 :CONF:EGPRs:BS:RLCM:RRBP? Value returned in this example: "N21"

:CONFigure:EGPRs:BS:RLCMac:RRBP:MODE

Syntax	:CONFigure:EGPRs:BS:RLCMac:RRBP:MODE <PredefExp1>
Parameters	PredefExp1 is one of the following predefined expressions: ON│OFF. Default is ON.
Description	This command indicates whether the value specified for the relative reserved block period (RRBP) shall be used by the mobile (ON) or not (OFF) in (asynchronous) generator/analyzer mode (please refer to the description of the :CONF:CSYS:GPG command for further information).
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	:CONFigure:EGPRs:BS:RLCMac:RRBP:MODE OFF :CONF:EGPRs:BS:RLCM:RRBP:MODE? Value returned in this example: "OFF"

:CONFigure:EGPRs:BS:RLCMac:PR

Syntax	:CONFigure:EGPRs:BS:RLCMac:PR <PredefExp1>
Parameters	PredefExp1 is one of the following predefined expressions: RNG6|RNG14|RNG22|RNG30. Default is RNG6.
Description	This command specifies the power reduction (PR) in (asynchronous) generator/analyzer mode (please refer to the description of the :CONF:CSYS:GPG command for further information). The power reduction the reduction in RF power (in dB) from the BCCH to the PDCH in downlink. PredefExp1 indicates the range of the power reduction: RNG6 stands for a power reduction of 0...6 dB, RNG14 indicates a power reduction of 8...14 dB, RNG22 stands for a power reduction of 16...22 dB, while RNG30 indicates a power reduction of 24...30 dB.
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	:CONFigure:EGPRs:BS:RLCMac:RRBP:PR RNG6 :CONF:EGPRs:BS:RLCM:RRBP:PR? Value returned in this example: "RNG6"

:CONFigure:EGPRs:BS:RLCMac:TFI

Syntax	:CONFigure:EGPRs:BS:RLCMac:TFI <int1>
Parameters	int1 is an integer. The minimum value for int1 is 0, the maximum value is 31. The default value for int1 is 0.
Description	This command sets the temporary flow identifier (TFI) in (asynchronous) generator/analyzer mode (please refer to the description of the :CONF:CSYS:GPG command for further information).
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:RLCMac:TFI 15 :CONF:EGPRs:BS:RLCM:TFI? Value returned in this example: "15"

:CONFigure:EGPRs:BS:RLCMac:BSN[:DATA]

Syntax	:CONFigure:EGPRs:BS:RLCMac:BSN[:DATA] <int1>
Parameters	int1 is an integer. The minimum value for int1 is 0, the maximum value is 127. The default value for int1 is 0.

Description	This command sets the block sequence number (BSN) in (asynchronous) generator/ analyzer mode (please refer to the description of the `:CONF:CSYS:GPG` command for further information).
	The BSN is the absolute block sequence number modulo 128 of each RLC data block within the TBF. It is used to enumerate the RLC data blocks, starting with 1.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	`:CONFigure:EGPRs:BS:RLCMac:BSN:DATA 1`
	`:CONF:EGPRs:BS:RLCM:BSN?`
	Value returned in this example: `"1"`

`:CONFigure:EGPRs:BS:RLCMac:BSN:INC`

Syntax	`:CONFigure:EGPRs:BS:RLCMac:BSN:INC <PredefExp1>`
Parameters	`PredefExp1` is one of the following predefined expressions: `ON`|`OFF`. Default is `ON`.
Description	This command switches the auto-increment of the block sequence number either on (`ON`) or off (`OFF`) as long as the Willtek 4400 is working in (asynchronous) generator/ analyzer mode (please refer to the description of the `:CONF:CSYS:GPG` command for further information).
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	`:CONFigure:EGPRs:BS:RLCMac:BSN:INC OFF`
	`:CONF:EGPRs:BS:RLCM:BSN:INC?`
	Value returned in this example: `"OFF"`

`:CONFigure:EGPRs:BS:ATTach`

Syntax	`:CONFigure:EGPRs: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 4400 when switched on or off, respectively. This can be used to detect when the mobile is switched on and listening to the Willtek 4400 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:EGPRs:BS:ATTach OFF`
	`:CONF:GSM:BS:ATT?`
	Value returned in this example: `"OFF"`.

:CONFigure:EGPRs:BS:MSTAtion:MSLot

Syntax	:CONFigure:EGPRs:BS:MSTAtion:MSLot <int1>
Parameters	int1 is an integer parameter. The minimum value for int1 is 1, the maximum value is 4. The default value for int1 is 1.
Description	This command either sets the number of UL time slots expected by the Willtek 4400 (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 4400. More details on multislot testing can be found in section Multislot Testing.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:MSTAtion:MSLot 4 :CONF:EGPRs:MSTA:MSL? Value returned in this example: "4".

:CONFigure:EGPRs:BS:BCC

Syntax	:CONFigure:EGPRs: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:BCC 1 :CONF:EGPRs:BS:BCC? Value returned in this example: "1".

:CONFigure:EGPRs:BS:BCH:ARFCn

Syntax	:CONFigure:EGPRs: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 4400's base channel (BCCH). **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:EGPRs:BS:BCH:ARFCn 60 :CONF:EGPRs:BS:BCH:ARFC? Value returned in this example: "60".

:CONFigure:EGPRs:BS:CBA

Syntax	`:CONFigure:EGPRs: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:EGPRs:BS:CBA 1`
	`:CONF:EGPRs:BS:CBA?`
	Value returned in this example: `"1"`.

:CONFigure:EGPRs:BS:CI

Syntax	`:CONFigure:EGPRs: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:EGPRs:BS:CI 127`
	`:CONF:EGPRs:BS:CI?`
	Value returned in this example: `"127"`.

:CONFigure:EGPRs:BS:LAI:MCC

Syntax	`:CONFigure:EGPRs: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:EGPRs:BS:LAI:MCC 238`
	`:CONF:EGPRs:BS:LAI:MCC?`
	Value returned in this example: `"238"`.

:CONFigure:EGPRs:BS:LAI:MNC

Syntax	:CONFigure:EGPRs:BS:LAI:MNC <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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:LAI:MNC 20 :CONF:EGPRs:BS:LAI:MNC? Value returned in this example: "20".

:CONFigure:EGPRs:BS:LAI:MNC:FORMat

Syntax	:CONFigure:EGPRs:BS:LAI:MNC:FORMat <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:EGPRs:BS:LAI:MNC:FORMat THRE :CONF:EGPRs:BS:LAI:MNC:FORM? Value returned in this example: "THRE".

:CONFigure:EGPRs:BS:LAI:LAC

Syntax	:CONFigure:EGPRs: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:LAI:LAC 17399 :CONF:EGPRs:BS:LAI:LAC? Value returned in this example: "17399".

:CONFigure:EGPRs:BS:LEVel

Syntax	:CONFigure:EGPRs:BS: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 resolution of real1 is 0.1. The default value for real1 is –60.0.
Description	Sets the (standard) output level of the Willtek 4400. If individual levels are selected for GPRS tests, this level is used on the base channel. Applies to GSM/GPRS synchronous mode.
Query	The query form of this command will return the current setting. The string delivered back will contain one real number.
Example	:CONFigure:EGPRs:BS:LEVel –100.0 :CONF:EGPRs:BS:LEVel? Value returned in this example: "–100.0".

:CONFigure:EGPRs:BS:MSLot[:DATA]

Syntax	:CONFigure:EGPRs:BS:MSLot[:DATA] <int1>
Parameters	int1 is an integer parameter. The minimum value for int1 is 1, the maximum value is 2. The default value for int1 is 1.
Description	Sets the number of PDTCH channels (slots) to be sent in the downlink. Applies to GPRS synchronous mode.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:MSLot[:DATA] 1 :CONF:EGPRs:BS:MSLot[:DATA]? Value returned in this example: "1".

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

Syntax	:CONFigure:EGPRs:BS:MSLot:LEVel[:DATA] <real1>, <real2>, <real3>, <real4>
Parameters	real1 to real4 are 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.
Description	Sets individual downlink slot levels for active (PDTCH) slots in Multislot GPRS operation (BCH remains at the level set by CONFigure:EGPRs:BS:LEVel. Applies to GPRS synchronous test mode. Data are used only if CONFigure:EGPRs:BS:MSLot:LEVel:MODE is set to INDL.
Query	The query form of this command will return the current setting. The string delivered back will contain four real numbers.
Example	:CONFigure:EGPRs:BS:MSLot:LEVel:DATA –80.0,–20.5,–60.0,–60.0 :CONF:EGPRs:BS:MSL:LEV:DATA? Values returned in this example: "–80.0,–20.5,–60.0,–60.0".

:CONFigure:EGPRs:BS:MSLot:LEVel:MODE

Syntax	:CONFigure:EGPRs:BS:MSLot:LEVel:MODE <PredefExp>
Parameters	PredefExp is one of the following predefined expressions: INDLev│STDLev. Default is INDLev.
Description	Selects between individual slot levels and standard slot level for PDTCHs. If STDLev is selected, the Willtek 4400 transmits at the level selected by CONFigure:EGPRs:BS:LEVel. If INDLev is selected, the Willtek 4400 uses the power levels selected with CONFigure:EGPRs:BS:MSLot:LEVel for the up to four used time slots; time slot 0 is transmitted at the level selected by CONFigure:EGPRs:BS:LEVel, and the level selected by CONFigure:EGPRs:BS:MSLot:UNUSed for the other slots.
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:EGPRs:BS:MSLot:LEVel:MODE INDLev :CONF:EGPRs:BS:MSL:LEV:MODE? Value returned in this example: "INDLev".

:CONFigure:EGPRs:BS:MSLot:LEVel:UNUSed

Syntax	:CONFigure:EGPRs:BS:MSLot:LEVel:UNUSed <real1>
Parameters	real1 is a floating point real number. The minimum value for real1 is –120.0, the maximum value is –10.0. The resolution of real1 is 0.1. The default value for real1 is –60.0.
Description	Sets the power level that the Willtek 4400 transmits in the unused time slots in packet data mode (i.e. except in time slot 0 and the assigned PDCH slots). Applies to GPRS synchronous mode.
Query	The query form of this command will return the current setting. The string delivered back will contain one real number.
Example	:CONFigure:EGPRs:BS:MSLot:LEVel:UNUSed –60.0 :CONF:EGPRs:BS:MSL:LEV:UNUS? Value returned in this example: "–60.0".

:CONFigure:EGPRs:BS:NCC

Syntax	:CONFigure:EGPRs: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:NCC 2 :CONF:EGPRs:BS:NCC? Value returned in this example: "2".

:CONFigure:EGPRs:BS:NCEL

Syntax	`:CONFigure:EGPRs:BS:NCEL1` `[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 4400. The mobile under test has to perform a MS report on those frequency channels. **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 4400 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 4400 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 neighbouring channel numbers currently in use. The string delivered back will always contain six integers.

Example	`:CONFigure:EGPRs:BS:NCEL1 10,20,30,40,50,60`
	`:CONF:EGPRs:BS:NCEL 70,80`
	`:CONF:EGPRs:BS:NCEL?`
	Values returned in this example: "`70,80,30,40,50,60`".

:CONFigure:EGPRs:BS:ORIGinate

Syntax	`:CONFigure:EGPRs:BS:ORIGinate <PredefExp>`
Parameters	PredefExp is one of the following predefined expressions: ONRequest|ONLOcation. Default is ONRequest.
Description	In GSM/GPRS, a TCH can be assigned either on request or as soon as the mobile has been registered (= immediately after a location update). `ONRequest` will assign a TCH only when either the mobile under test or the Willtek 4400 initializes a call. `ONLOcation` will assign a TCH as soon as the registration has been completed.
Query	The query form of this command will return the current setting. The string delivered back will contain the short-form version one of the predefined expressions as explained above.

Example	`:CONFigure:EGPRs:BS:ORIGinate ONLOcation`
	`:CONF:EGPRs:BS:ORIG?`
	Value returned in this example: "`ONLO`".

:CONFigure:EGPRs:BS:PDTCh:ARFCn

Syntax	:CONFigure:EGPRs:BS:PDTCh: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 2.
Description	This command sets the frequency channel number of the simulated packet data traffic channel.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:PDTCh 2 :CONF:EGPRs:BS:PDTC? Value returned in this example: "2".

:CONFigure:EGPRs:BS:TCH:ARFCn

Syntax	:CONFigure:EGPRs: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:TCH:ARFCn 917 :CONF:EGPRs:BS:TCH:ARFC? Value returned in this example: "917".

:CONFigure:EGPRs:BS:TCH:TSLot

Syntax	:CONFigure:EGPRs: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).
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.
Example	:CONFigure:EGPRs:BS:TCH:TSLot 5 :CONF:EGPRs:BS:TCH:TSL? Value returned in this example: "5".

:CONFigure:EGPRs:BS:TCH:TYPE

Syntax	:CONFigure:EGPRs: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.
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:EGPRs:BS:TCH:TYPE EFR :CONF:EGPRs:BS:TCH:TYPE? Value returned in this example: "EFR".

:CONFigure:EGPRs:BER:RTDelay

Syntax	:CONFigure:EGPRs: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:EGPRs:BER:RTDelay 9 :CONF:EGPRs:BER:RTD? Value returned in this example: "9"

:CONFigure:EGPRs:MSTAtion:DRX

Syntax	:CONFigure:EGPRs: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). **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:EGPRs:MSTAtion:DRX 2`
	`:CONF:EGPRs:MSTA:DRX?`
	Value returned in this example: "2", resulting in a DRX period of four multiframes.

:CONFigure:EGPRs:MSTAtion:TADVance

Syntax	`:CONFigure:EGPRs: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.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:EGPRs:MSTAtion:TADVance 12`
	`:CONF:EGPRs:MSTA:TADV?`
	Value returned in this example: "12".

:CONFigure:EGPRs:MSTAtion:PLEVel[:ALL]

Syntax	`:CONFigure:EGPRs: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:EGPRs:MSTAtion:PLEVel:ALL 17`
	`:CONF:EGPRs:MSTA:PLEV?`
	Value returned in this example: "17".

:CONFigure:EGPRs:MSTAtion:MODE

Syntax	`:CONFigure:EGPRs: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 signalling channels in GSM: SDCCH and FACCH. The FACCH is always associated with a traffic channel. However, some mobiles will only listen to signalling provided on the SDCCH.

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:EGPRs:BS:CMODe FACCh`
	`:CONF:EGPRs:BS:CMOD?`
	Value returned in this example: "FACC".

:CONFigure:EGPRs:MSTAtion:MSLot

Syntax	`:CONFigure:EGPRs: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 4400 (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 the Willtek 4400.
Query	The query form of this command will return the current setting. The string delivered back will contain one integer.

Example	`:CONFigure:EGPRs:MSTAtion:MSLot 4`
	`:CONF:EGPRs:MSTA:MSL?`
	Value returned in this example: "4".

:CONFigure:EGPRs:MSTAtion:LOOPback

Syntax	`:CONFigure:EGPRs:MSTAtion:LOOPback <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `ON` | `OFF`. Default is `ON`.
Description	Selects the loopback mode. If the parameter `PredefExp` is set to `ON`, the mobile station will support GPRS loopback (mode B), if the parameter `PredefExp` is set to `OFF`, the mobile station will not support GPRS loopback (mode A).
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:EGPRs:MSTAtion:LOOPback ON`
	`:CONF:EGPRs:MSTA:LOOP?`
	Value returned in this example: "ON".

:CONFigure:EGPRs:MSTAtion:PSLots

Syntax	`:CONFigure:EGPRs:MSTAtion:PSLots <int1>, <int2>,` `<int3>, <int4>`
Parameters	`int1` to `int4` are integer parameters. The minimum value for each `intx` is –26, the maximum value is 39. The resolution of each `intx` is 2, the default value for each `intx` is 5.

Description	Sets the uplink power level in the range of –23 … +39 dBm (GSM900) or –26 … +36 dBm (upper band). Applies to GPRS synchronous mode.
Query	The query form of this command will return the current setting. The string delivered back will contain four integers.

Example	`:CONFigure:EGPRs:MSTAtion:PSLots -10,-6,0,6`
	`:CONF:EGPRs:MSTA:PSL?`
	Value returned in this example: "`-10,-6,0,6`".

`:CONFigure:EGPRs:TYPE`

Syntax	`:CONFigure:EGPRs: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 4400 was brought into GSM call mode. `GSM9001800` will make the Willtek 4400 work as a test set for the GSM system types GSM 900 and GSM 1800. `GSM9001900` will set up the Willtek 4400 as a test set for the GSM system types GSM 900 and GSM 1900. Notes – This command will have no effect when the Willtek 4400 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 regions.
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:EGPRs:TYPE GSM9001900`
	`:CONF:EGPRs:TYPE?`
	Value returned in this example: "`GSM9001900`".

`:CONFigure:EGPRs:MEASure:GROUp[:RFTX]`

Syntax	`:CONFigure:EGPRs: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 you to set a user-specific sequence of RF TX tests on the Willtek 4400. The sequence has to be started with the appropriate commands of the MEASure subsystem (`:MEAS:EGPRs:CONT:RFTX:GROU` or `:MEAS:EGPRs: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 pre-defined positions of a burst.
	`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 4400. The string delivered back will contain a maximum of nine short-form versions of the predefined expressions explained above
Example	`:CONFigure:EGPRs:MEASure:GROUp:RFTX PRMS,POWer,FLATness`
	`:CONF:EGPRs:MEAS:GROU?`
	Value returned in this example: `"PRMS,POW,FLAT"`

`:CONFigure:EGPRs:MEASure:GROUp:AFANalyser`

Syntax	`:CONFigure:EGPRs: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 you to set a user-specific sequence of AF tests on the Willtek 4400. 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 the Willtek 4400.
Query	The query form of this command will return the current sequence of AF tests to be performed by the Willtek 4400. The string delivered back will contain a maximum of six predefined short-form versions of the expressions explained above.

Example	`:CONFigure:EGPRs:MEASure:GROUp:AFANalyser SINad,DISTortion,FREQuency`
	 `:CONF:EGPRs:MEAS:GROU:AFAN?`
	 Value returned in this example: `"SIN,DIST,FREQ"`

`:CONFigure:EGPRs:MEASure:ACPM:TRANsient`

Syntax	`:CONFigure:EGPRs: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 4400 calculates the RF output spectrum due to switching transients (or ACP Transient for short). If `FULL` is selected, the Willtek 4400 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:EGPRs:MEASure:ACPM:TRAN FULL`
	 `:CONF:EGPRs:MEASure:ACPM:TRAN?`
	 In this example, the query will return the string `FULL`.

:CONFigure:EGPRs:MSLot:RESult

Syntax	:CONFigure:EGPRs:MSLot:RESult <int1>
Parameters	int1 is an integer parameter. The minimum value for int1 is 1, the maximum value is 4. The default value for int1 is 1.
Description	Sets the number of the burst for which to deliver measurement results. Delivers the results for the selected slot out of the number of all measured slots. This command can be used in GPRS synchronous mode when the mobile transmits on multiple time slots but results are fetched from one (selectable) time slot only.
Query	The query form of this command is not available.
Example	:CONFigure:EGPRs:MSLot:RESult 1 :CONF:EGPRs:MSLot:RESult? Value returned in this example: "1".

CALL subsystem

This subsystem contains commands

–   for call setup and handling procedures and

–   to read out the measurement report, generated by the mobile.

:CALL:GPRS:BSORiginate

Syntax	:CALL:GPRS:BSORiginate
Parameters	There are no parameters.
Description	Initiates a call setup by the base station (Willtek 4400). Before initiating a call, ensure that the GPRS Attach procedure has been completed to avoid any confusion in the call procedures. See the **:STATus:OPERation:SIGNalling:GPRS[:EVENt]** command for more details.
Query	There is no query form of this command available.
Example	:CALL:GPRS:BSOR This command will initiate a BS-originated call.

:CALL:GPRS:RSIGnaling

Syntax	:CALL:GPRS:RSIGnaling
Parameters	There are no parameters.
Description	Makes the Willtek 4400 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. For details regarding reduced signaling, please refer to section Reduced Signaling.

Query	There is no query form of this command available.
Example	:CALL:GPRS:RSIG

:CALL:GPRS:BSRelease

Syntax	:CALL:GPRS:BSRelease
Parameters	There are no parameters.
Description	This command will make the Willtek 4400 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:GPRS:BSR This command will release the call.

:CALL:GPRS:PAGing

Syntax	:CALL:GPRS: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 4400 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:GPRS:PAG ON This command switches continuous paging on.

:CALL:GPRS:ASSignment

Syntax	:CALL:GPRS: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:GPRS:ASS EARL Early assignment will be used.

:CALL:GPRS:BERLoop

Syntax	:CALL:GPRS: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 4400 will tell the mobile under test to close its internal loop, while OFF will make the Willtek 4400 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:GPRS:BERL ON :CALL:GPRS:BERL? Value returned: "ON".

:CALL:GPRS:MSINfo:ALL

Syntax	:CALL:GPRS: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. **RX Qual (Integer)** Based on the number of bits corrected by its own channel decoder, the mobile estimates the BER. **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. **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. **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. **Dualband (Integer)** A 6 on this position indicates that the mobile under test supports both the E-GSM and the GSM 1800 bands. **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:GPRS:MSINfo:ALL?`
	The string delivered back is:
	`"49,0,"+4989996410","001010123456789",`
	`"520040190000430",4,2,49,1,0,3,0,1,16,1".`

:CALL:GPRS:MSINfo:RXLevel

Syntax	`:CALL:GPRS:MSINfo:RXLevel`
Parameters	There are no parameters.
Description	The mobile estimates the RF power level at which it receives the base station.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	`:CALL:GPRS:MSINfo:RXL?` The string delivered back is: `"49"`.

:CALL:GPRS:MSINfo:RXQual

Syntax	`:CALL:GPRS: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.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	`:CALL:GPRS:MSINfo:RXQ?` The string delivered back is: `"0"`.

:CALL:GPRS:MSINfo:NUMBer

Syntax	`:CALL:GPRS: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:GPRS:MSINfo:NUMB?` The string delivered back is: `"+4989996410"`.

:CALL:GPRS:MSINfo:IMSI

Syntax	`:CALL:GPRS:MSINfo:IMSI`
Parameters	There are no parameters.

Description	This query will deliver back 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:GPRS:MSINfo:IMSI?` The string delivered back is: `"001010123456789"`.

:CALL:GPRS:MSINfo:IMEI

Syntax	`:CALL:GPRS: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:GPRS:MSINfo:IMSI?` The string delivered back is: `"520040190000430"`.

:CALL:GPRS:MSINfo:MSCLass

Syntax	`:CALL:GPRS:MSINfo:MSCLass`
Parameters	There are no parameters.
Description	This query will deliver back the code representing the mobile's power class.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	`:CALL:GPRS:MSINfo:MSCLass?` The integer delivered back is: `"4"`.

:CALL:GPRS:MSINfo:RLEVel

Syntax	`:CALL:GPRS: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:GPRS:MSINfo:RLEVel?` The integer delivered back is: `"2"`.

:CALL:GPRS:MSINfo:RXLSub

Syntax	`:CALL[:GPRS):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.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	:CALL:GPRS:MSINfo:RXLSub? The integer delivered back is: "49".

:CALL:GPRS:MSINfo:SMS

Syntax	:CALL:GPRS: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:GPRS:MSINfo:SMS? The boolean number delivered back is: "1".

:CALL:GPRS:MSINfo:EGSM

Syntax	:CALL:GPRS: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:GPRS:MSINfo:EGSM? The boolean number delivered back is: "0".

:CALL:GPRS:MSINfo:AFIVe

Syntax	:CALL:GPRS: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:GPRS:MSINfo:AFIVe? The integer delivered back is: "3".

:CALL:GPRS:MSINfo:TADVance

Syntax	:CALL:GPRS:MSINfo:TADVance
Parameters	There are no parameters.
Description	This query will deliver back the timing advance currently used by the mobile (in bit periods).
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	:CALL:GPRS:MSINfo:TADVance? The integer delivered back is: "0".

:CALL:GPRS:MSINfo:DUALband

Syntax	:CALL:GPRS: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).
Query	There is only a query form of this command available. The query will deliver back one integer indicating the supported bands: 1 - P-GSM only 2 - E-GSM only 4 - GSM 1800 only 5 - P-GSM and GSM 1800 6 - E-GSM and GSM 1800
Example	:CALL:GPRS:MSINfo:DUALband? The integer delivered back is: "6".

:CALL:GPRS:MSINfo:PCONtrol

Syntax	:CALL:GPRS:MSINfo:PCONtrol
Parameters	There are no parameters.
Description	This query will deliver back the power control level currently used by the mobile.
Query	There is only a query form of this command available. The query will deliver back one integer.
Example	:CALL:GPRS:MSINfo:PCONtrol? The integer delivered back is: "16".

:CALL:GPRS:MSINfo:EFRCapab

Syntax	:CALL:GPRS: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:GPRS:MSINfo:EFRCapab?` The boolean number delivered back is: `"1"`.

:CALL:GPRS:MSINfo:ATYPe

Syntax	`:CALL:GPRS:MSINfo:ATYPe?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	This query returns a string containing an integer number pointing to the attach type. A return value of 1 indicates that the mobile requested a standalone GPRS attach. 2 indicates that the mobile requested a GPRS attach after IMSI attach. 3 means that the mobile requested a combined IMSI/GPRS attach.
Example	`:CALL:GPRS:MSINfo:ATYPe?` Returns a string like the following: `"2"`

:CALL:GPRS:MSINfo:MCLass:GPRS

Syntax	`:CALL:GPRS:MSINfo:MCLass:GPRS?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Returns a string containing the GPRS multislot class reported by the MS. Valid return values are in the range 1 to 29.
Example	`:CALL:GPRS:MSIN:MCL:GPRS?` Returns a string, for example `"3"`

:CALL:GPRS:MSINfo:MCLass:HSCSd

Syntax	`:CALL:GPRS:MSINfo:MCLass:HSCSd?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Returns a string containing the HSCSD multislot class reported by the MS. Valid return values are in the range 1 to 18.
Example	`:CALL:GPRS:MSIN:MCL:HSCS?` Returns a string, for example `"2"`

:CALL:GPRS:MSINfo:SMTime

Syntax	`:CALL:GPRS:MSINfo:SMTime?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Returns a string containing the switch-measure time reported by the mobile phone. Valid return values are in the range 0 through 15.
Example	`:CALL:GPRS:MSIN:SMT?` Returns a string, for example 4

:CALL:GPRS:MSINfo:SMSTime

Syntax	`:CALL:GPRS:MSINfo:SMSTime?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Returns a string containing the switch-measure-switch time reported by the mobile phone. Valid return values are in the range 0 through 15.
Example	`:CALL:GPRS:MSIN:SMST?` Returns a string, for example 4

:CALL:GPRS:MSINfo:ATTached?

Syntax	`:CALL:GPRS: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 4400. 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. The GSM attach is not to be confused with the GPRS attach.
Example	`:CALL:GPRS:MSINfo:ATT?` The boolean number delivered back is: `"1"`.

:CALL:GPRS:MODE

Syntax	`:CALL:GPRS:MODE <PredefExpr>`
Parameters	`PredefExp` is one of the following expressions: `TX` | `BCS` | `USF` | `STOP` | `DETach`.

Description	Initiates the signaling state required for the respective measurement, or leads back to idle mode.
	`TX` starts the signaling required for transmitter measurements.
	`BCS` starts signaling to bring the mobile into a state where BLER-BCS measurements are possible.
	`USF` initiates signaling that allows both BLER-USF and TX measurements.
	`STOP` initiates signaling to terminate the current uplink and/or downlink TBF.
	`DETach` starts the signaling to resume GPRS idle mode.
Query	Returns the short form predefined expression for MODE.
Example	`:CONF:GPRS:MSTA:MSL 3`
	`CALL:GPRS:MODE USF`
	`MEAS:GPRS:MULT:TX:ALL?`Initiates an uplink TBF on three time slots, starts the transmitter measurements and returns results for three time slots.

`:CALL:EGPRs:MSINfo:ALL`

Syntax	`:CALL:EGPRs:MSINfo:ALL`
Parameters	There are no parameters.
Description	This command reads out all EGPRS MS Info result values with one query.
	The query will deliver back 3 single values in the following sequence, all separated by commas:
	– IMSI (String) This is the international mobile subscriber identity.
	– IMEI (String) This position contains the international mobile equipment identity.
	– MClass shows the GPRS multislot class reported by the MS. Valid return values are in the range 1 to 29.
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:EGPRs:MSINfo:ALL?`
	The string delivered back in this example is: `"001010123456789"`, `"520040190000430"`,3.

`:CALL:EGPRs:MSINfo:IMSI`

Syntax	`:CALL:EGPRs:MSINfo:IMSI`
Parameters	There are no parameters.
Description	This query will deliver back 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:EGPRs:MSINfo:IMSI?`
	The string delivered back in this example is: `"001010123456789"`.

`:CALL:EGPRs:MSINfo:IMEI`

Syntax	`:CALL:EGPRs: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:EGPRs:MSINfo:IMEI?` The string delivered back in this example is: `"520040190000430"`.

:CALL:EGPRS:MSINfo:MCLass:GPRS

Syntax	`:CALL:EGPRS:MSINfo:MCLass:GPRS?`
Parameters	There are no parameters.
Description	Only the query form is supported.
Query	Returns a string containing the GPRS multislot class reported by the MS. Valid return values are in the range 1 to 29.
Example	`CALL:EGPRS:MSIN:MCL:GPRS?` Returns a string, for example `"10"`

:CALL:EGPRs:MODE

Syntax	`:CALL:EGPRs:MODE <PredefExpr>`
Parameters	`PredefExp` is one of the following expressions: `TX`|`STOP`|`DETach`.
Description	Initiates the signaling state required for the respective measurement, or leads back to idle mode. `TX` starts the signaling required for transmitter measurements. `STOP` initiates signaling to terminate the current uplink and/or downlink TBF. `DETach` starts the signaling to resume GPRS idle mode.
Query	Returns the short form predefined expression for MODE.

Example	`:CONF:EGPRs:MSTA:MSL 3`
	`CALL:EGPRs:MODE TX`
	`MEAS:EGPRs:MULT:TX:ALL?`
	Initiates a downlink TBF on three time slots, starts the transmitter measurements and returns results for three time slots.

MEASure subsystem

The MEASure subsystem is probably the most important SCPI command subsystem of the 4400. There, you will find all commands required to acquire measurement results of the mobile under test.

Please note that the accuracy specified for multislot transmitter measurements applies to the time slot with the highest power level. The accuracy of results at levels of more than 10 dB below the highest power level may be significantly lower.

: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 (see section "The MEASure subsystem" on page 365 for details). – To perform any audio measurements on your Willtek 4400, the Audio Option must be installed. – For further details regarding group measurements, please refer to section ":MEASure:…:GROup" on page 372.
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 (see section "The MEASure subsystem" on page 365 for details). – Any AF measurement will need the Audio Option to be installed on your Willtek 4400. – For further details regarding group measurements, please refer to section ":MEASure:…:GROup" on page 372.
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 4400.
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 4400.
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 4400.
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 4400.
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 4400.
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 4400.
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 4400, 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 4400. – 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 4400.
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 4400.
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 4400.
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: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 (see section "The MEASure subsystem" on page 365 for details). – To perform any audio measurements on your Willtek 4400, the Audio Option must be installed. – For further details regarding group measurements, please refer to section ":MEASure:…:GROup" on page 372. – More information regarding **ARRay** measurements can be found in section "MEASure:ARRay" on page 368.
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 (see section "The MEASure subsystem" on page 365 for details). – Any AF measurement will need the Audio Option to be installed on your Willtek 4400. – For further details regarding group measurements, please refer to section ":MEASure:…:GROup" on page 372.
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 4400.

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 measurent 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: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 4400.
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 4400.
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 4400.
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 4400.
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 4400, the MS Power Supply and Current Measurements Option must be installed. – For further details regarding group measurements, please refer to section ":MEASure:...:GROup" on page 372.
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. **Notes** – Any power and current consumption measurement will need the MS Power Supply and the Current Measurement Options to be installed on your Willtek 4400. – For further details regarding group measurements, please refer to section ":MEASure:...:GROup" on page 372.
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:GPRS[:CONTinuous]:RFTX:GROup

Syntax	`:MEASure:GPRS[:CONTinuous]:RFTX:GROup`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the GSM RF TX tests, specified with the `:CONF:GPRS:MEAS:GRO:RFTX` command. To read out the latest measurement results, use the `:FETCh:GPRS: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:GPRS: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:GPRS:CONT:RFTX:ALL` command.

Example	`:CONF:GPRS:MEAS:GRO:RFTX POW,PRMS`
	`:MEASure:GPRS: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:GPRS[:CONTinuous]:RFTX:ALL

Syntax	`:MEASure:GPRS[:CONTinuous]:RFTX:ALL`
Parameters	There are no parameters.

Description	Starts a continuous measurement of the most important GSM RF TX tests. To read out the latest measurement results, use the `:FETCh:GPRS: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 `[:CONTinuous]` measurements in general, check with section `:MEASure:GPRS[:CONTinuous]`. 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:GPRS: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:GPRS[:CONTinuous]:RFTX:PPEAk

Syntax	:MEASure:GPRS[:CONTinuous]:RFTX:PPEAk
Parameters	There are no parameters.
Description	Starts a continuous measurement of the GSM peak phase error. To read out the latest measurement result, use the :FETCh:GPRS: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:GPRS:CONTinuous:RFTX:PPEAk? The string returned is: "5.13".

:MEASure:GPRS[:CONTinuous]:RFTX:PRMS

Syntax	:MEASure:GPRS[:CONTinuous]:RFTX:PRMS
Parameters	There are no parameters.
Description	Starts a continuous measurement of the root-mean square valued (RMS) phase error in GSM. To read out the latest measurement result, use the :FETCh:GPRS: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:GPRS:CONTinuous:RFTX:PRMS? The string returned is: "1.54".

:MEASure:GPRS[:CONTinuous]:RFTX:FREQuency

Syntax	:MEASure:GPRS[:CONTinuous]:RFTX:FREQuency
Parameters	There are no parameters.
Description	Starts a continuous measurement of the mobile's frequency error in GSM mode. To read out the latest measurement result, use the :FETCh:GPRS: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:GPRS:CONTinuous:RFTX:FREQuency? The string returned is: "-31.92".

:MEASure:GPRS[:CONTinuous]:RFTX:LENGth

Syntax	:MEASure:GPRS[:CONTinuous]:RFTX:LENGth
Parameters	There are no parameters.
Description	Starts a continuous measurement of the burst length in GSM mode. To read out the latest measurement result, use the :FETCh:GPRS: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:GPRS[:CONTinuous]:RFTX:UTIMe

Syntax	:MEASure:GPRS[:CONTinuous]:RFTX:UTIMe
Parameters	There are no parameters.
Description	Starts a continuous measurement of the GSM uplink timing error. To read out the latest measurement result, use the :FETCh:GPRS: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:GPRS:CONTinuous:RFTX:UTIMe? The string returned is: "0.1".

:MEASure:GPRS[:CONTinuous]:RFTX:POWer

Syntax	:MEASure:GPRS[: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:GPRS: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:GPRS:CONTinuous:RFTX:POWer? The string returned is: "11.22".

:MEASure:GPRS[:CONT]:RFTX:TEMPlate

Syntax	:MEASure:GPRS[: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:GPRS: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:GPRS:CONTinuous:RFTX:TEMPlate This command will start the continuous check of the burst against the PTT.

:MEASure:GPRS[:CONT]:RFTX:CORNer

Syntax	:MEASure:GPRS[: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:GPRS: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:GPRS:CONTinuous:RFTX:CORNer? The string returned is: "-72.18,-61.91,-20.91,-0.05, -0.04,-17.97,-56.60,-73.95".

:MEASure:GPRS[:CONT]:RFTX:FLATness

Syntax	:MEASure:GPRS[: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:GPRS: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:GPRS:CONTinuous:RFTX:FLATness?` The string returned is: `"-0.12,113,7,0.56,34.0"`.

:MEASure:GPRS[:CONT]:RFTX:FPOWer

Syntax	`:MEASure:GPRS[: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:GPRS:RFTX:POWer` measurement, but it is faster. To read out the latest measurement result, use the `:FETCh:GPRS:RFTX:FPOWer` command. For further details, refer to the description of the `:CONF:GPRS: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:GPRS:CONTinuous:RFTX:FPOWer?` The string returned is: `"13.05"`.

:MEAS:GPRS[:CONT]:BLOC:BURStshape

Syntax	`:MEAS:GPRS[:CONT]:BLOC: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:GPRS:BLOCkdata:BURStshape` command. The aim of this measurement is to obtain the pure RF power levels, sampled by the Willtek 4400 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:GPRS: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 in this example.
	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 4400 during the burst period.

`:MEAS:GPRS[:CONT]:BLOC:PHASeerror`

Syntax	`:MEAS:GPRS[:CONT]:BLOC: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:GPRS:BLOCkdata:PHASeerror` command.
	The aim of this measurement is to obtain the pure phase error levels, calculated by the Willtek 4400 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:GPRS:CONT:BLOCkdata:PHASeerror?` The array returned in this example 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.

:MEAS:GPRS[:CONT]:BLOC:MSP[:CURR]

Syntax	`:MEAS:GPRS[:CONT]:BLOC:MSP[:CURR]`
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:GPRS:BLOCkdata:MSPectrum` command. The aim of this measurement is to obtain the pure modulation spectrum levels, calculated by the Willtek 4400 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:GPRS:CONT:BLOCkdata:MSPectrum?` The array returned is:`"-62.0,-61.5,-62.2, ...-0.4,0.0,-0.3,...-62.4"`.

:MEAS:GPRS[:CONT]:BLOC:MSP:AVG

Syntax	`:MEAS:GPRS[:CONT]:BLOC:MSP: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:GPRS: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:GPRS:CONT:BLOCkdata:MSPectrum:AVG? 25` The array returned is: `"-61.0,-61.0,-61.1, ...0.0,0.0,0.0,...-61.0"`.

`:MEASure:GPRS[:CONT]:RFRX:RBER:ALL`

Syntax	`:MEASure:GPRS[: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:GPRS: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:GPRS: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:GPRS: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:GPRS[:CONT]:RFRX:RBER:CIB

Syntax	`:MEASure:GPRS[: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:GPRS: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:GPRS: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:GPRS[:CONT]:RFRX:RBER:CII

Syntax	`:MEASure:GPRS[: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:GPRS: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:GPRS: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:GPRS:CONT:RFRX:RBER:CII?`
	Measurement result string delivered back:
	`"1.2".`

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

Syntax	`:MEASure:GPRS[: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:GPRS: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:GPRS: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:GPRS:CONT:RFRX:RBER:FER?` Measurement result string delivered back: `"0.0"`.

`:MEASure:GPRS[:CONT]:RFRX:BER:ALL`

Syntax	`:MEASure:GPRS[: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:GPRS: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:GPRS: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:GPRS: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:GPRS[:CONT]:RFRX:BER:CIA

Syntax	:MEASure:GPRS[: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:GPRS: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:GPRS: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:GPRS:CONT:RFRX:BER:CIA? Measurement result string delivered back: "0.0".

:MEASure:GPRS[:CONT]:RFRX:BER:CIB

Syntax	:MEASure:GPRS[: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:GPRS: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:GPRS: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:GPRS:CONT:RFRX:BER:CIB? Measurement result string delivered back: "0.1".

:MEASure:GPRS[:CONT]:RFRX:BER:CII

Syntax	:MEASure:GPRS[: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:GPRS: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:GPRS: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:GPRS:CONT:RFRX:BER:CII?` Measurement result string delivered back: `"0.2"`.

:MEASure:GPRS[:CONTinuous]:RFRX:BER:CRC

Syntax	`:MEASure:GPRS[:CONTinuous]:RFRX:BER:CRC`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the (non-residual) bit error ratio on the class II bits. To read out the latest measurement result, use the `:FETCh:GPRS: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:GPRS: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:GPRS:RFRX:BER:CRC?` Measurement result string delivered back in this example: `"0.2"`.

:MEASure:GPRS[:CONTinuous]:RFRX:FBER

Syntax	`:MEASure:GPRS[: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:GPRS: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:GPRS: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:GPRS:CONT:RFRX:FBER?` Measurement result string delivered back: `"1.5"`.

`:MEASure:GPRS[:CONT]:RFRX:RTDelay`

Syntax	`:MEASure:GPRS[: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:GPRS: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:GPRS:CONT:RFRX:RTDelay?` Measurement result string delivered back: `"8"`.

`:MEAS:GPRS[:CONT]:RFSP:ACPM:MOD`

Syntax	`:MEAS:GPRS[:CONT]:RFSP:ACPM:MOD`
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:GPRS:RFSP:ACPM:MOD` command. **Note:** According to GSM specification 05.05, this measurement will be performed on ($\pm$)100 kHz and ($\pm$)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 ($\pm$)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:GPRS: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:GPRS[:CONTinuous]:RFSPectrum:ACPM:MODPower

Syntax	:MEASure:GPRS[: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:GPRS: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. 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 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). 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:GPRS: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:GPRS[:CONT]:RFSP:ACPM:TRANsient

Syntax	:MEAS:GPRS[: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:GPRS:RFSP:ACPM:TRANsient? 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:GPRS: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:GPRS[:CONTinuous]:MULTislot:RFTX:ALL`

Syntax	`:MEASure:GPRS[:CONTinuous]:MULTislot:RFTX:ALL`
Parameters	There are no parameters.
Description	Measures repetitively all RF TX parameters for the number of time slots set by the `:CONFigure:GPRS:MSTAtion:MSLot` command. The number of time slots measured is the first parameter in the result string. To read out the latest measurement results, use the `:FETCh:GPRS:MULTislot: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` is 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 by returning 1 and 0 that the power/time template has and has not been violated by a burst respectively, `CORNer` gives the RF power level in dBm at eight predefined positions of a burst, `FLATness` indicates 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 more information on the single measurements, see description of the related commands below.
Query	The query form of this command starts the measurements and - after all measurements have been completed and all measurement results have been obtained for all time slots - delivers a string, containing an integer number indicating the number of time slots measured followed by 19 measured values for each time slot, separated with commas. The order and type of the measurement results delivered back are as follows: 1. The number of time slots measured, an integer number, 2. `PPEAk`, floating point real number, 3. `PRMS`, floating point real number, 4. `FREQuency`, floating point real number, 5. `LENGth`, floating point real number, 6. `UTIMe`, floating point real number, 7. `POWer`, floating point real number, 8. `TEMPlate`, a boolean number, 9...16. `CORNer`, eight floating point real numbers, 17...20. `FLATness`, four floating point real numbers. The items 2... 20 are repeated in the same order and format for each time slot measured.

Example	`:MEAS:GPRS:CONT:MULT:RFTX:ALL?`
	For the number of measured time slots set to 1, 19 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 the number of time slots measured and the measurement results. Example:
	`"1,5.13,1.94,-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:GPRS[:CONTinuous]:MULTislot:RFTX:PPEAk

Syntax	`:MEASure:GPRS[:CONTinuous]:MULTislot:RFTX:PPEAk`
Parameters	There are no parameters.
Description	Measures repetitively the peak phase error for all time slots assigned by the Willtek 4400 using the `:CONFigure:GPRS:MSTAtion:MSLot` command. To read out the latest measurement result, use the `:FETCh:GPRS:MULTi-slot:RFTX:PPEAk` command.
Query	The query form of this command starts the measurements and returns measurement results as soon as the measurements have been completed. The string delivered back contains an integer number indicating the number of time slots measured followed by floating point real numbers with the physical dimension of degrees, representing the measurement results for each time slot.

Example	`:MEAS:GPRS:CONT:MULT:RFTX:PPEA?`
	For the number of time slots measured set to 2, an example of the string returned is:
	`"2,5.13,6.28".`

:MEASure:GPRS[:CONTinuous]
:MULTislot:RFTX:TEMPlate

Syntax	`:MEASure:GPRS[:CONTinuous]:MULTislot:RFTX:TEMPlate`
Parameters	There are no parameters.
Description	Checks repetitively the TX bursts against the power/time template (PTT). To read out the latest measurement result, use the `:FETCh:GPRS:MULTislot:RFTX:TEMPlate` command. **Note:** The power/time templates are user-definable and can be set up using the `:CALC:GPRS:RFTX:TEMP` commands.
Query	The query form of this command starts the check and returns measurement results as soon as the check has been completed. The string delivered back contains an integer number indicating the number of time slots measured, followed by boolean numbers signaling in turn possible violation of the PTT for the rising edge, for the first slot (active part), for the first transition period, ...for the last burst (active part), and for the falling edge. A 0 indicates **no** violation of the PTT, while a 1 means that a burst measured violated the PTT.
Example	`:MEAS:GPRS:CONT:MULT:RFTX:TEMP?` For two TX bursts measured, an example of the string returned is: `"2,0,1,0,0,0"` indicating a PTT violation in the active part of the first burst.

`:MEASure:GPRS[:CONTinuous]`
`:MULTislot:RFSPectrum:ACPM:MODulation`

Syntax	`:MEASure:GPRS[:CONTinuous]:MULTislot:RFSPec-` `trum:ACPM:MODulation`
Parameters	There are no parameters.
Description	Measures repetitively the output spectrum due to modulation for each of the time slots assigned by the Willtek 4400 using the :CONFigure:GPRS:MSTAtion:MSLot command. To read out the latest measurement results, use the `:FETCh:GPRS:MULT: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 measurements and delivers a string, containing an integer indicating the number of time slots measured followed by 23 floating point real numbers for each time slot representing the spectrum values with the physical dimension dBc (dB relative to the carrier, being the transmission frequency of the mobile), 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:GPRS:CONT:MULT:RFSP:ACPM:MOD?`
	For the number of time slots measured set to 1, an example of the string returned is:
	`"1,5.1,6.3,10.2,12.8,9.6,16.0,15.3,18.9,17.5,13.3,12.2,`
	`11.9,10.6,9.1,8.7,7.6,9.6,6.4,5.9,4.4,6.7,3.6,2.7"`.

`:MEASure:GPRS[:CONTinuous]`
`:MULTislot:RFSPectrum:ACPM:TRANsient`

Syntax	`:MEASure:GPRS[:CONTinuous]:MULTislot:RFSPec-` `trum:ACPM:TRANsient`
Parameters	There are no parameters.
Description	Measures repetitively the output spectrum due to switching transients for each of the time slots assigned by the Willtek 4400 using the `:CONFigure:GPRS:MSTA-tion:MSLot` command. To read out the latest measurement results, use the `:FETCh:GPRS: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 measurements and delivers a string, containing an integer indicating the number of time slots measured followed by 27 floating point real numbers for each time slot representing the spectrum values with the physical dimension dBc (dB relative to the carrier, being the transmission frequency of the mobile), separated by commas.

Example	`:MEAS:GPRS:CONT:MULT:RFSP:ACPM:TRANS?`
	For the number of time slots measured set to 1, an example of the string returned is:
	`"1,5.1,6.3,10.2,12.8,9.6,16.0,15.3,18.9,17.5,13.3,12.2,`
	`11.9,10.6,9.1,8.7,7.6,9.6,6.4,5.9,4.4,6.7,3.6,2.7,3.3,4`
	`.1,1.9,2.5".`

`:MEASure:GPRS[:CONTinuous]:MULTislot:RFRX:BLER`

Syntax	`:MEASure:GPRS[:CONTinuous]:MULTislot:RFRX:BLER`
Parameters	There are no parameters.
Description	Measures repetitively the BLER in USF or BLER-BCS test mode for each of the time slots assigned by the Willtek 4400 using the `:CONFigure:GPRS:MSTAtion:MSLot` and `:CONFigure:GPRS:BS:MSLot[:DATA]` commands respectively. The test mode can be set by the `:CALL:GPRS:MODE` command. **Notes:** 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:GPRS:BLER:COUNt` command. To read out the latest measurement results, use the `:FETCh:GPRS:MULTislot:RFRX:BLER` 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 an integer number indicating the number of time slots measured, a floating point real number giving the BLER for all slots followed by floating point real numbers giving the BLER for individual slots, separated by commas.

Example	`:MEASure:GPRS:CONT:MULT:RFRX:BLER?`
	For the number of time slots set to 2, an example of the string returned is:
	`"2,0.2,0.0,0.2".`

`:MEASure:GPRS:ARRay:RFTX:GROup`

Syntax	`:MEASure:GPRS: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:GPRS: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:GPRS: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: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:GPRS:CONT:RFTX:ALL` command. **Note:** Please keep in mind that the number of measurement results delivered back is the number of measurement results of a single execution multiplied with the `int1` parameter.

Example	`:CONF:GPRS:MEAS:GRO:RFTX POW,PRMS`
	`:MEASure:GPRS: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) measured value delivered back is the RMS phase error, while the second (fourth and sixth) measured value is the mobile's RF output power level.

:MEASure:GPRS:ARRay:RFTX:ALL

Syntax	`:MEASure:GPRS: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	Measures multiple times (set with the `int1` parameter) all RF TX parameters. To read out the entire measurement results array, use the `:FETCh:GPRS: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` is 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 by returning 1 and 0 that the power/time template has and has not been violated by a burst respectively, `CORNer` gives the RF power level in dBm at eight predefined positions of a burst, `FLATness` indicates 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 more information on the single measurements, see description of the related commands below.

Query	The query form of this command returns a string, containing `int1` times 19 measured values separated with commas. The order and type of the measurement results delivered back are as follows: 1. `PPEAk`, floating point real number, 2. `PRMS`, floating point real number, 3. `FREQuency`, floating point real number, 4. `LENGth`, floating point real number, 5. `UTIMe`, floating point real number, 6. `POWer`, floating point real number, 7. `TEMPlate`, a boolean number, 8...15. `CORNer`, eight floating point real numbers, 16...19. `FLATness`, four floating point real numbers. The items 1... 19 are repeated in the same order and format `int1` times for all measurements.

Example	`:MEAS:GPRS:ARR:RFTX:ALL 2`
	`:FETCH:GPRS:RFTX:ALL?`
	In this example, the sequence of the RX TX measurements will be performed twice. All 38 measured 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: Consequently, a subsequent `:FETCh` command (without any previous start of a measurement) will try to read nonexisting measurement data and result in a time-out error.

`:MEASure:GPRS:ARRay:RFTX:PPEAk`

Syntax	`:MEASure:GPRS: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	Measures multiple times (set with the `int1` parameter) the peak phase error. To read out the entire measurement results array, use the `:FETCh:GPRS:RFTX:PPEAk` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of degrees, representing the measurement results. The single measurement results are separated by commas.
Example	`:MEAS:GPRS:ARR:RFTX:PPEA? 5` An example of the string returned is: `"5.13,5.23,4.97,5.12,4.59"`.

`:MEASure:GPRS:ARRay:RFTX:PRMS`

Syntax	`:MEASure:GPRS: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	Measures multiple times (set with the `int1` parameter) the rms phase error. To read out the entire measurement results array, use the `:FETCh:GPRS:RFTX:PRMS` command.

Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of degrees, representing the measurement results. The single measurement results are separated by commas.
Example	`:MEAS:GPRS:ARR:RFTX:PRMS? 5` An example of the string returned is: `"5.13,5.23,4.97,5.12,4.59"`.

:MEASure:GPRS:ARRay:RFTX:FREQuency

Syntax	`:MEASure:GPRS: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	Measures multiple times (set with the `int1` parameter) the frequency error. To read out the entire measurement results array, use the `:FETCh:GPRS:RFTX:FREQ` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of Hz, representing the measurement results. The single measurement results are separated by commas.

Example	`:MEAS:GPRS:ARR:RFTX:FREQ 5`
	`:FETC:GPRS:RFTX:FREQ?`
	The frequency error measurement will be performed five times. The measurement results will be stored internally and read by the `FETC` command. An example of the string returned is: `"22.33,22.53,21.97,22.41,21.79"`.

:MEASure:GPRS:ARRay:RFTX:LENGth

Syntax	`:MEASure:GPRS: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	Measures multiple times (set with the `int1` parameter) the burst length. To read out the entire measurement results array, use the `:FETCh:GPRS:RFTX:LENGth` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of microseconds, representing the measurement results. The single measurement results are separated by commas.

Example	`:MEAS:GPRS:ARR:RFTX:LENG 5`
	`:FETC:GPRS:RFTX:LENG?`
	The measurement of the burst length will be performed five times. The measurement results will be stored internally and read by the `FETC` command. An example of the string returned is: `"556.4,557.2,558.2,557.8,556.5"`.

:MEASure:GPRS:ARRay:RFTX:UTIMe

Syntax	`:MEASure:GPRS: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	Measures multiple times (set with the `int1` parameter) the uplink timing error. To read out the entire measurement results array, use the `:FETCh:GPRS:RFTX:UTIMe` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of microseconds, representing the measurement results. The single measurement results are separated by commas.
Example	`:MEAS:GPRS:ARR:RFTX:UTIM? 5` An example of the string returned is: `"0.0,0.1,0.0,-0.2,0.1"`.

:MEASure:GPRS:ARRay:RFTX:POWer

Syntax	`:MEASure:GPRS:ARRay:RFTX:POWer <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	Measures multiple times (set with the `int1` parameter) the burst peak power. To read out the entire measurement results array, use the `:FETCh:GPRS:RFTX:POWer` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of dBm, representing the measurement results. The single measurement results are separated by commas.
Example	`:MEAS:GPRS:ARR:RFTX:POW? 5` An example of the string returned is: `"11.22,11.09,11.21,11.14,10.99"`.

:MEASure:GPRS:ARRay:RFTX:TEMPlate

Syntax	`:MEASure:GPRS: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	Checks multiple times (set with the `int1` parameter) if TX burst fits into the power/time template. To read out the entire measurement results array, use the `:FETCh:GPRS:RFTX:TEMPlate` command. **Note:** The power/time template is user-definable. It can be set up using the `:CALC:GPRS:RFTX:TEMP` commands.
Query	The query form of this command returns a string, containing `int1` boolean numbers. A 0 indicates no violation of the PTT by the current burst, while a 1 signals PTT violation. The single measurement results are separated by commas.

Example	`:MEAS:GPRS:ARR:RFTX:TEMP 10`
	`:FETC:GPRS:RFTX:TEMP?`
	The check of the current burst against the PTT will be performed ten times. The check resuls will be stored internally and read out by the `FETC` command. An example of the string returned is: `"0,0,0,0,0,0,1,0,0,0"`, indicating that the seventh burst checked violated the PTT.

:MEASure:GPRS:ARRay:RFTX:CORNer

Syntax	`:MEASure:GPRS: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	Measures multiple times (set with the `int1` parameter) the RF power levels at eight user-definable positions specified by the `CALC:GPRS:RFTX:CORN:POS` command. To read out the entire measurement results array, use the `:FETCh:GPRS:RFTX:CORNer` command.
Query	The query form of this command returns a string, containing `int1` times 8 floating point real numbers with the physical dimension of dBm, representing the measurement results. The single measurement results are separated by commas.
Example	`:MEAS:GPRS:ARR:RFTX:CORN? 2` An example of 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:GPRS:ARRay:RFTX:FLATness

Syntax	`:MEASure:GPRS: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	Measures multiple times (set with the `int1` parameter) the flatness of the burst top level. 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. 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:GPRS:RFTX:FLATness` command.
Query	The query form of this command returns a string, containing (4 * `int1`) floating point real numbers with the physical dimensions as explained above. The single measurement results are separated by commas.

Example	`:MEASure:GPRS: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:GPRS:ARRay:RFTX:FPOWer

Syntax	`:MEASure:GPRS:ARRay:RFTX:FPOWer <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	Measures multiple times (set with the `int1` parameter) the fast burst peak power. This measurement is very similar to the `:MEAS:GPRS:ARR:RFTX:POW` measurement, but its is faster. To read out the entire measurement results array, use the `:FETCh:GPRS:RFTX:FPOWer` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of dBm, representing the measurement results. The single measurement results are separated by commas.
Example	`:MEAS:GPRS:ARR:RFTX:FPOW? 5` An example of the string returned is: `"13.02,13.09,13.01,13.14,13.29"`.

:MEASure:GPRS:ARRay:RFRX:RBER:ALL

Syntax	`:MEASure:GPRS: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:GPRS: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:GPRS:BER:COUNt` command. For further details regarding group measurements, please refer to section `:MEASure:...:GROup`. More information regarding `ARRay` measurements can be found in section `:MEASure:ARRay`.

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:GPRS:ARRay:RFRX:RBER:ALL 2`
	`:FETCh:GPRS: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:GPRS:ARRay:RFRX:RBER:CIB

Syntax	`:MEASure:GPRS: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:GPRS: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:GPRS: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:GPRS:ARRay:RFRX:RBER:CIB? 5` The string returned is: `"0.2,0.2,0.1,0.3,0.2"`.

:MEASure:GPRS:ARRay:RFRX:RBER:CII

Syntax	`:MEASure:GPRS: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:GPRS: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:GPRS: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:GPRS:ARRay:RFRX:RBER:CII? 5` The string returned is: `"1.4,1.4,1.5,1.7,1.3"`.

`:MEASure:GPRS:ARRay:RFRX:RBER:FER`

Syntax	`:MEASure:GPRS: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:GPRS: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:GPRS: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:GPRS:ARRay:RFRX:RBER:FER 5` `:FETCh:GPRS:RFRX:RBER:FER?` The string returned is: `"0.0,0.1,0.0,0.2,0.1"`.

`:MEASure:GPRS:ARRay:RFRX:BER:ALL`

Syntax	`:MEASure:GPRS: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:GPRS: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 `":CONFigure:GPRS:BLER:COUNt"` command.
	For further details regarding group measurements, please refer to section `":MEASure:...:GROup"`.
	More information regarding `ARRay` measurements can be found in section `"MEASure:ARRay"`.
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:GPRS:ARRay:RFRX:BER:ALL 2`
	`:FETCh:GPRS: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:GPRS:ARRay:RFRX:BER:CIA

Syntax	`:MEASure:GPRS: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:GPRS: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:GPRS: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:GPRS:ARRay:RFRX:BER:CIA? 5` The string returned is: `"1.4,1.4,1.5,1.7,1.3"`.

:MEASure:GPRS:ARRay:RFRX:BER:CIB

Syntax	`:MEASure:GPRS: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:GPRS: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:GPRS: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:GPRS:ARRay:RFRX:BER:CIB? 5` The string returned is: `"1.4,1.4,1.5,1.7,1.3"`.

:MEASure:GPRS:ARRay:RFRX:BER:CII

Syntax	`:MEASure:GPRS: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:GPRS: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:GPRS: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:GPRS:ARRay:RFRX:BER:CII? 5 The string returned is: "1.4,1.4,1.5,1.7,1.3".

:MEASure:GPRS:ARRay:RFRX:BER:CRC

Syntax	:MEASure:GPRS:ARRay:RFRX:BER:CRC <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 (non-residual) 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:GPRS: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:GPRS: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	:MEAS:GPRS:ARR:RFRX:BER:CRC? 5 The string returned in this example is: "1.4,1.4,1.5,1.7,1.3".

:MEASure:GPRS:ARRay:RFRX:FBER

Syntax	:MEASure:GPRS: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:GPRS: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:GPRS: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:GPRS:ARRay:RFRX:FBER? 5 The string returned is: "1.4,1.4,1.5,1.7,1.3".

:MEASure:GPRS:ARRay:RFRX:RTDelay

Syntax	:MEASure:GPRS: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:GPRS: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:GPRS:ARRay:RFRX:RTDelay? 5 The string returned is: "8,8,8,8,8".

:MEAS:GPRS:ARR:RFSP:ACPM:MODulation

Syntax	:MEAS:GPRS: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:GPRS:RFSP:ACPM:MOD command. **Note:** According to GSM specification 05.05, this measurement will be performed on ($\pm$)100 kHz and ($\pm$)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 ($\pm$)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:GPRS:ARR:RFSPectrum:ACPM:MODulation 5 :FETCh:GPRS:RFSP:ACPM:MOD? The string returned would contain 115 floating point real numbers, representing the 23 measurement result values of five measurement runs.

:MEASure:GPRS:ARRay:RFSPectrum:ACPM:MODPower

Syntax	:MEASure:GPRS: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:GPRS: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:GPRS:ARR:RFSP:ACPM:MODP 5 :FETCh:GPRS: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:GPRS:ARR:RFSP:ACPM:TRANsient

Syntax	:MEAS:GPRS: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:GPRS: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. 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 (27 * `int1`) floating point real numbers. The physical dimension is dBm. The single measurement results are separated by commas.

Example	`:MEAS:GPRS:ARR:RFSPectrum:ACPM:TRANsient 5`
	`:FETCh:GPRS:RFSP:ACPM:TRAN?`
	The string returned would contain 135 floating point real numbers, representing the 27 measurement result values of five measurement runs.

:MEASure:GPRS:ARRay:MULTislot:RFTX:ALL

Syntax	`:MEASure:GPRS:ARRay:MULTislot:RFTX:ALL <int1>`
Parameters	`int1` specifies how many MULT:RFTX:ALL measurements (over multiple slots of a frame) shall be performed. Up to 100 measurements can be specified.
Description	Measures all RFTX parameters for a defined number of times, on all the available uplink slots.
Query	Performs `int1` times an RFTX:ALL measurement on all the available uplink slots and returns a string containing `int1` times the result string from the `:MEASure:GPRS:CONT:MULT:RFTX:ALL` query.

Example	`:CONF:GPRS:MSTA:MSL 2`
	`:MEAS:GPRS:ARR:MULT:RFTX:ALL 10`
	`:FETCh:GPRS:MULT:RFTX:ALL?`
	Delivers a string containing 10*(1+2*19)=390 results, separated by commas.

:MEASure:GPRS:ARRay:MULTislot:RFTX:PPEAk

Syntax	`:MEASure:GPRS:ARRay:MULTislot:RFTX:PPEAk <int1>`
Parameters	`int1` specifies how many MULT:RFTX:PPEAk measurements (over multiple slots of a frame) shall be performed. Up to 100 measurements can be specified.
Description	Measures `int1` times the RFTX parameter peak phase error, on all the available uplink slots.
Query	Performs `int1` times a peak phase error measurement on all the available uplink slots and returns a string containing `int1` times the result string from the `:MEASure:GPRS:CONT:MULT:RFTX:PPEAk` query.

Example	`:CONF:GPRS:MSTA:MSL 2`
	`:MEAS:GPRS:ARR:MULT:RFTX:PPEAk? 3`
	Returns a string containing nine results separated by commas, for example
	`2,8.3,9.6, 2,8.1,8.9, 2,8.2,9.5`

:MEASure:GPRS:ARRay:MULTislot:RFTX:TEMPlate

Syntax	`:MEASure:GPRS:ARRay:MULTislot:RFTX:TEMPlate <int1>`
Parameters	`int1` specifies how many measurements shall be taken. Up to 100 measurements can be specified.

Description	Checks repetitively whether the `int1` TX bursts fit into the respective power/time templates.
Query	The return string contains `int1` times the number of the uplink time slots followed by the pass/fail results for the starting edge, the active part of the first slot, the transition period between the first and the second slot, the active part of the second slot and so on until the last active part and the trailing edge. A 0 indicates that the burst shape is inside limits, 1 indicates outside limits for the respective part.

Example	`:CONF:GPRS:MSTA:MSL 2`
	`:MEAS:GPRS:ARR:MULT:RFTX:TEMP? 3`
	Returns a string like the following:
	`2,0,0,0,0,0, 2,0,0,1,0,0, 2,0,0,0,1,0`
	In this example, the signal hit the limits during the transition period in the second measurement and during the active part of the second slot in the third measurement.

:MEASure:GPRS:ARRay
:MULTislot:RFSPectrum:ACPM:MODulation

Syntax	`:MEASure:GPRS:ARRay:MULTislot:RFSPectrum:ACPM:MODulation <int1>`
Parameters	`int1` specifies how many measurements shall be taken. Up to 100 measurements are supported.
Description	Measures a defined number of times the RF output spectrum due to modulation on all the active uplink slots.
Query	Starts the measurement and returns int1 times the number of active uplink slots followed by 23 float values per time slot. For the meaning of the 23 float values, see :MEAS:GPRS:ACPM:MODulation.
Example	`:CONF:GPRS:MSTA:MSL 2`

:MEASure:GPRS:ARRay
:MULTislot:RFSPectrum:ACPM:TRANsient

Syntax	`:MEASure:GPRS:ARRay:MULTislot:RFSPectrum:ACPM:TRANsient <int1>`
Parameters	`int1` specifies how many measurements shall be taken. Up to 100 measurements are supported.
Description	Measures a defined number of times the RF output spectrum due to switching transients on all the active uplink slots.
Query	Starts the measurement and returns `int1` times the number of active uplink slots followed by 27 float values per time slot. For the meaning of the 27 float values, see `:MEAS:GPRS:ACPM:TRANsient`.
Example	`:CONF:GPRS:MSTA:MSL 2`

`:MEASure:GPRS:ARRay:MULTislot:RFRX:BLER`

Syntax	`:MEASure:GPRS:ARRay:MULTislot:RFRX:BLER`
Parameters	`int1` specifies how many measurements shall be taken. Up to 100 measurements are supported.
Description	Measures a specified number of times the block error rate (BLER) for all the available slots. The results can be requested with the `:FETCh:GPRS:MULT:RFRX:BLER` command. **Note:** Two types of BLER are available: BLER-USF and BLER-BCS. The type is selected with the `:CALL:GPRS:MODE` command.
Query	Starts `int1` measurements and returns a string containing `int1` times the number of active slots followed by the block error rate (BLER) on each of these time slots. The BLER is expressed in % and a floating point real value.

Example	`:CONF:GPRS:MSTA:MSL 2`
	 `:CALL:GPRS:MODE USF`
	 `:MEAS:GPRS:ARR:MULT:RFRX:BLER? 3`
	 Returns a string like
	 `2,0.7,1.4, 2,0.9,1.5, 2,0.8,1.3`

`:MEASure:EGPRs[:CONTinuous]:RFTX:GROup`

Syntax	`:MEASure:EGPRs[:CONTinuous]:RFTX:GROup`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the GSM RF TX tests, specified with the `:CONF:EGPRs:MEAS:GRO:RFTX` command. To read out the latest measurement results, use the `:FETCh:EGPRs:RFTX:GROup` command. **Notes:** – Please keep in mind that the start of a new RF TX test will always terminate the preceding one instantly. – For further details regarding group measurements and their specific advantages, refer to section `:MEAS:...:GROup`. – For [:CONTinuous] measurements in general, check with section `:MEASure:EGPRs[:CONTinuous]`.
Query	The query form of this command will start the sequence of RF TX measurements as specified with the `:CONF:EGPRs: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:EGPRs:CONT:RFTX:ALL` command.

Example	`:CONF:EGPRs:MEAS:GRO:RFTX POW,EVM`
	 `:MEASure:EGPRs:CONTinuous:RFTX:GROup?`
	 In this example, the group of measurements is defined by a power level measurement combined with an error vector magnitude measurement. The string returned in this example is: "`4.53,0.05`".
	 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:EGPRs[:CONTinuous]:RFTX:ALL

Syntax	:MEASure:EGPRs[:CONTinuous]:RFTX:ALL
Parameters	There are no parameters.
Description	Starts a continuous measurement of the most important EGPRS RF TX tests. To read out the latest measurement results, use the :FETCh:EGPRs:RFTX:ALL command. The RF TX tests performed by this command are:
	ERMS, the error vector magnitude rms-averaged over a burst.
	EVM is a relative value and shown as a percentage.
	EPEAk, the maximum error vector magnitude within a burst.
	EVM is indicated in per cent.
	ENFTh, the error vector magnitude that is not exceeded by 95 % of the symbols of a burst. The 95 % EVM is shown in %.
	EOFFset, the origin offset indicates the accuracy of the i/Q modulators DC setup; it is usually an undesire leakage produced due to a DC offset in the phones I/Q modulator. It is measured in dBc, 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).
	Notes: Please keep in mind that the start of a new RF TX test will always terminate the preceding one instantly. For further details regarding group measurements and their specific advantages, refer to section :MEAS:...:GROup. For [:CONTinuous] measurements in general, check with section :MEASure:EGPRs[:CONTinuous]. 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 measurements have been completed and all measurement results obtained - delivers a string, containing 17 measurement result values, separated with commas. The order and type of these measurement result values delivered back is as follows: 1. ERMS, the error vector magnitude rms-averaged over a burst. EVM is a relative value and shown as a percentage, 2. EPEAk, the maximum error vector magnitude within a burst. EVM is indicated per cent, 3. ENFTh, the error vector magnitude that is not exceeded by 95% of the symbols of a burst. The 95% EVM is shown in %, 4. EOFFset, the origin offset indicates the accuracy of the i/Q modulators DC setup; it is usually an undesire leakage produced due to a DC offset in the phones I/Q modulator. It is measured in dBc, 5. FREQuency, floating point real number representing the mobile's frequency error, 6. LENGth, floating point real number representing the length of the burst in microseconds, 7. UTIMe, floating point real number representing the timing error of the mobile under test in microseconds (taking the ordered timing advance into account), 8. POWer, floating point real number representing the mobile's RF output power level in dBm, 9. 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. 10...17. CORNer, eight floating point real numbers, representing the RF power level in dBm at eight predefined positions of a burst.

Example	`:MEASure:EGPRs:CONTinuous:RFTX:ALL?`
	In this case, all relevant EGPRS 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 17 measurement results:
	`"5.13,11.94,2.43,4.55,-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".`

`:MEASure:EGPRs[:CONTinuous]:RFTX:ERMS`

Syntax	`:MEASure:EGPRs[:CONTinuous]:RFTX:ERMS`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the error vector magnitude rms-averaged over a burst. EVM is a relative value and shown as a percentage. To read out the measurement results, use the `:FETCh:EGPRs:RFTX:ERMS` 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 dimension of %.

Example	`:MEAS:EGPRs:RFTX:ERMS`
	`:FETC:EGPRs:RFTX:ERMS?`
	The error vector magnitude rms-averaged measurement will be performed. The measurement result will be stored internally and read by the <TT>FETC</TT> command. An example of the string returned is: `"2.33"`.

:MEASure:EGPRs[:CONTinuous]:RFTX:EPEAk

Syntax	:MEASure:EGPRs[:CONTinuous]:RFTX:EPEAk
Parameters	There are no parameters.
Description	Starts a continuous measurement of the maximum error vector magnitude within a burst. EVM is indicated per cent. To read out the measurement results, use the :FETCh:EGPRs:RFTX:EPEAk 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 dimension of %.
Example	:MEAS:EGPRs:RFTX:EPEAk? The maximum error vector magnitude measurement will be performed and the result will be returned. Example of the returned string: "6.33".

:MEASure:EGPRs[:CONTinuous]:RFTX:ENFTh

Syntax	:MEASure:EGPRs[:CONTinuous]:RFTX:ENFTh
Parameters	There are no parameters.
Description	Starts a continuous measurement of the error vector magnitude that is not exceede by 95% of the symbols of a burst. The 95% EVM is shown in %. To read out the measurement results, use the :FETCh:EGPRs:RFTX:ENFTh 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 dimension of %.
Example	:MEAS:EGPRs:RFTX:ENFTh :FETC:EGPRs:RFTX:ENFTh? The maximum error vector magnitude measurement will be performed. The measurement result will be stored internally and read by the FETC command. An example of the string returned is: "4.33".

:MEASure:EGPRs[:CONTinuous]:RFTX:EOFFset

Syntax	:MEASure:EGPRs[:CONTinuous]:RFTX:EOFFset
Parameters	There are no parameters.
Description	Starts a continuous measurement of the origin offset. The origin offset indicates the accuracy of the i/Q modulators DC setup; it is usually an undesire leakage produced due to a DC offset in the phones I/Q modulator. It is measured in dBc. To read out the measurement results, use the :FETCh:EGPRs:RFTX:ENFTh 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 dimension of dBc.

Example	`:MEAS:EGPRs:FTX:EOFFset?` The maximum error vector magnitude measurement will be performed and the result will be returned, for example: `"1.97"`.

:MEASure:EGPRs[:CONTinuous]:RFTX:FREQuency

Syntax	`:MEASure:EGPRs[:CONTinuous]:RFTX:FREQuency`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the mobile's frequency error in GSM mode. To read out the latest measurement result, use the `:FETCh:EGPRs: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:EGPRs:CONTinuous:RFTX:FREQuency?` The string returned in this example is: `"-31.92"`.

:MEASure:EGPRs[:CONTinuous]:RFTX:LENGth

Syntax	`:MEASure:EGPRs[:CONTinuous]:RFTX:LENGth`
Parameters	There are no parameters.
Description	Starts a continuous measurement of the burst length in GSM mode. To read out the latest measurement result, use the `:FETCh:EGPRs: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:EGPRs:CONTinuous:RFTX:LENGth?` The string returned in this example is: `"557.0"`.

:MEASure:EGPRs[:CONTinuous]:RFTX:UTIMe

Syntax	`:MEASure:EGPRs[:CONTinuous]:RFTX:UTIMe`
Parameters	There are no parameters
Description	Starts a continuous measurement of the GSM uplink timing error. To read out the latest measurement result, use the `:FETCh:EGPRs: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:EGPRs:CONTinuous:RFTX:UTIMe? The string returned in this example is: "0.1".

:MEASure:EGPRs[:CONTinuous]:RFTX:POWer

Syntax	:MEASure:EGPRs[: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:EGPRs: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:EGPRs:CONTinuous:RFTX:POWer? The string returned in this example is: "11.22".

:MEASure:EGPRs[:CONT]:RFTX:TEMPlate

Syntax	:MEASure:EGPRs[: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:EGPRs: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:EGPRs:CONTinuous:RFTX:TEMPlate This command will start the continuous check of the burst against the PTT.

:MEASure:EGPRs[:CONT]:RFTX:CORNer

Syntax	:MEASure:EGPRs[: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:EGPRs: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:EGPRs:CONTinuous:RFTX:CORNer?`
	The string returned in this example is: `"-72.18,-61.91,-20.91,-0.05, -0.04,-17.97,-56.60,-73.95"`.

`:MEAS:EGPRs[:CONT]:BLOC:BURStshape`

Syntax	`:MEAS:EGPRs[:CONT]:BLOC: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:EGPRs:RFTX:BLOCkdata:BURStshape` command.
	The aim of this measurement is to obtain the pure RF power levels, sampled by the Willtek 4400 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.
	Further information regarding this kind of measurement can be found in section `:MEASure:...:BLOCkdata`.
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:EGPRs:CONT:BLOC:BURStshape?`
	The array returned in this example 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 in this example.
	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 4400 during the burst period.

:MEAS:EGPRs[:CONT]:BLOC:MSP[:CURR]

Syntax	`:MEAS:EGPRs[:CONT]:BLOC:MSP[:CURR]`
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:EGPRs:RFTX:BLOCkdata:MSPectrum` command.
	The aim of this measurement is to obtain the pure modulation spectrum levels, calculated by the Willtek 4400 in order to graphically display the modulation spectrum curve on another computer or to perform some special analysis of the modulation spectrum.
	Further information regarding this kind of measurement can also be found in section `:MEASure:...:BLOCkdata`.
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:EGPRs:CONT:BLOCkdata:MSPectrum?`
	The array returned in this example is: `"-62.0,-61.5,-62.2, ...-0.4,0.0,-0.3,...-62.4"`.

:MEAS:EGPRs[:CONT]:BLOC:MSP:AVG

Syntax	`:MEAS:EGPRs[:CONT]:BLOC:MSP: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. Further information regarding this kind of measurement can be found in section `:MEASure:...:BLOCkdata`. **Note:** To read out the measurement result values of the averaging process, the `:FETCh:EGPRs:RFTX:BLOCkdata:MSPectrum` command can 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:EGPRs:CONT:BLOCkdata:MSPectrum:AVG? 25` The array returned in this example is: "`-61.0,-61.0,-61.1, ...0.0,0.0,0.0,...-61.0`".

:MEASure:EGPRs[:CONTinuous]:RFSPectrum:ACPM: MODulation

Syntax	`:MEASure:EGPRs[:CONTinuous]:RFSPectrum: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:EGPRs:RFSP:ACPM:MOD` command. **Note:** According to GSM specification 05.05, this measurement will be performed on ($\pm$)100 kHz and ($\pm$)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 ($\pm$)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:EGPRs: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:EGPRs[:CONTinuous]:RFSPectrum:ACPM: MODPower

Syntax	`:MEASure:EGPRs[: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:EGPRs: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. 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 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). 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:EGPRs:CONT: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:EGPRs[:CONT]:RFSP:ACPM:TRANsient

Syntax	`:MEAS:EGPRs[: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:EGPRs: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 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:EGPRs: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:EGPRs[:CONTinuous]:MULTislot:RFTX:TEMPlat

Syntax	`:MEASure:EGPRs[:CONTinuous]:MULTislot:RFTX:TEMPlate`
Parameters	There are no parameters.
Description	Checks repetitively the TX bursts against the power/time template (PTT). To read out the latest measurement result, use the `:FETCh:EGPRs:MULTislot:RFTX:TEMPlate` command. **Note:** The power/time templates are user-definable and can be set up using the `:CALC:EGPRs:RFTX:TEMP` commands.
Query	The query form of this command starts the check and returns measurement results as soon as the check has been completed. The string delivered back contains an integer number indicating the number of time slots measured, followed by boolean numbers signaling in turn compliance with or violation of the PTT for the each individual slot. A 0 indicates no violation of the PTT, while a 1 means that a burst measured violated the PTT.
Example	`:MEAS:EGPRs:CONT:MULT:RFTX:TEMP?` For two TX bursts measured, an example of the string returned is: "2,1,0" indicating a PTT violation in the first slot.

**:MEASure:EGPRs[:CONTinuous]:MULTislot:RFSPectrum:
ACPM:MODulatio**

Syntax	`:MEASure:EGPRs[:CONTinuous]:MULTislot:RFSPectrum:ACPM:MODulation`
Parameters	There are no parameters.

Description	Measures repetitively the output spectrum due to modulation for each of the time slots assigned by the Willtek 4400 using the `:CONFigure:EGPRs:MSTA-tion:MSLot` command. To read out the latest measurement results, use the `:FETCh:EGPRs:MULT: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 measurements and delivers a string, containing an integer indicating the number of time slots measured followed by 23 floating point real numbers for each time slot representing the spectrum values with the physical dimension dBc (dB relative to the carrier, being the transmission frequency of the mobile), 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:EGPRs:CONT:MULT:RFSP:ACPM:MOD?`
	For the number of time slots measured set to 1, an example of the string returned is:
	`"1,5.1,6.3,10.2,12.8,9.6,16.0,15.3,18.9,17.5,13.3,12.2,`
	`11.9,10.6,9.1,8.7,7.6,  9.6,6.4,5.9,4.4,6.7,3.6,2.7".`

`:MEASure:EGPRs[:CONTinuous]:MULTislot:RFSPectrum:ACPM:TRANsient`

Syntax	`:MEASure:EGPRs[:CONTinuous]:MULTislot:RFSPectrum:ACPM:TRANsient`
Parameters	There are no parameters.
Description	Measures repetitively the output spectrum due to switching transients for each of the time slots assigned by the Willtek 4400 using the :CONFigure:EGPRs:MSTAtion:MSLot command. To read out the latest measurement results, use the :FETCh:EGPRs: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 measurements and delivers a string, containing an integer indicating the number of time slots measured followed by 27 floating point real numbers for each time slot representing the spectrum values with the physical dimension dBc (dB relative to the carrier, being the transmission frequency of the mobile), separated by commas.

Example	`:MEAS:EGPRs:CONT:MULT:RFSP:ACPM:TRANS?`
	For the number of time slots measured set to 1, an example of the string returned is:
	`"1,5.1,6.3,10.2,12.8,9.6,16.0,15.3,18.9,17.5,13.3,12.2,`
	`11.9,10.6,9.1,8.7,7.6,9.6,6.4,5.9,4.4,6.7,3.6,2.7,3.3,`
	`4.1,1.9,2.5".`

:MEASure:EGPRs:ARRay:RFTX:GROup

Syntax	:MEASure:EGPRs: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:EGPRs: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:EGPRs: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: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:EGPRs:CONT:RFTX:ALL command. **Note:** Please keep in mind that the number of measurement results delivered back is the number of measurement results of a single execution multiplied with the int1 parameter.
Example	:CONF:EGPRs:MEAS:GRO:RFTX POW,ERMS :MEASure:EGPRs: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 in this example 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) measured value delivered back is the RMS phase error, while the second (fourth and sixth) measured value is the mobile's RF output power level.

:MEASure:EGPRs:ARRay:RFTX:ALL

Syntax	:MEASure:EGPRs:ARRay:RFTX:ALL <int1&gt>
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 EGPRS 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:EGPRs:RFTX:ALL` command.
	The RF TX tests performed by this command are:
	`ERMS`, the error vector magnitude rms-averaged over a burst. EVM is a relative value and shown as a percentage,
	`EPEAk`, the maximum error vector magnitude within a burst. EVM is indicated per cent,
	`ENFTh`, the error vector magnitude that is not exceede by 95% of the symbols of a burst. The 95% EVM is shown in %,
	`EOFFset`, the origin offset indicates the accuracy of the i/Q modulators DC setup; it is usually an undesire leakage produced due to a DC offset in the phones I/Q modulator. It is measured in dBc,
	`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 pre-defined positions of a burst (see section Corner Points for details).
	Notes: Please keep in mind that the start of a new RF TX test will always terminate the preceding one instantly. For further details regarding group measurements and their specific advantages, refer to section `:MEAS:...:GROup`.
	For `ARRay` measurements in general, check with section `:MEASure:...:ARRay`. 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 (17 * `<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. `ERMS`, the error vector magnitude rms-averaged over a burst. EVM is a relative value and shown as a percentage,
	2. `EPEAk`, the maximum error vector magnitude within a burst. EVM is indicated per cent,
	3. `ENFTh`, the error vector magnitude that is not exceede by 95% of the symbols of a burst. The 95% EVM is shown in %,
	4. `EOFFset`, the origin offset indicates the accuracy of the i/Q modulators DC setup; it is usually an undesire leakage produced due to a DC offset in the phones I/Q modulator. It is measured in dBc,
	5. `FREQuency`, floating point real number representing the mobile's frequency error,
	6. `LENGth`, floating point real number representing the length of the burst in microseconds,
	7. `UTIMe`, floating point real number representing the timing error of the mobile under test in microseconds (taking the ordered timing advance into account),
	8. `POWer`, floating point real number representing the mobile's RF output power level in dBm,
	9. `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.
	10...17. `CORNer`, eight floating point real numbers, representing the RF power level in dBm at eight predefined positions of a burst.

Example	`:MEASure:EGPRs:ARRay:RFTX:ALL 2`
	`:FETCh:EGPRs:RFTX:ALL?`
	In this case, the sequence of the most important EGPRS RF TX measurements will be performed twice. All 34 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 non-existing measurement data and thus create a timeout error.

:MEASure:EGPRs:ARRay:RFTX:ERMS

Syntax	`:MEASure:EGPRs:ARRay:RFTX:ERMS <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	Measures multiple times (set with the `int1` parameter) the error vector magnitude rms-averaged over a burst. EVM is a relative value and shown as a percentage. To read out the entire measurement results array, use the `:FETCh:EGPRs:RFTX:ERMS` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of %, representing the measurement results. The single measurement results are separated by commas.

Example	`:MEAS:EGPRs:ARR:RFTX:ERMS 5`
	`:FETC:EGPRs:RFTX:ERMS?`
	The error vector magnitude rms-averaged measurement will be performed five times. The measurement results will be stored internally and read by the `FETC` command. An example of the string returned is: `"2.33,2.53,1.97,2.41,1.79"`.

:MEASure:EGPRs:ARRay:RFTX:EPEAk

Syntax	`:MEASure:EGPRs:ARRay:RFTX:EPEAk <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	Measures multiple times (set with the `int1` parameter) the maximum error vector magnitude within a burst. EVM is indicated per cent. To read out the entire measurement results array, use the `:FETCh:EGPRs:RFTX:EPEAk` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of %, representing the measurement results. The single measurement results are separated by commas.

Example	`:MEAS:EGPRs:ARR:RFTX:EPEAk 5`
	`:FETC:EGPRs:RFTX:EPEAk?`
	The maximum error vector magnitude measurement will be performed five times. The measurement results will be stored internally and read by the `FETC` command. An example of the string returned is: `"6.33,8.53,11.97,9.41,10.79"`.

:MEASure:EGPRs:ARRay:RFTX:ENFTh

Syntax	`:MEASure:EGPRs:ARRay:RFTX:ENFTh <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	Measures multiple times (set with the `<int1>` parameter) the error vector magnitude that is not exceeded by 95% of the symbols of a burst. The 95% EVM is shown in %. To read out the entire measurement results array, use the `:FETCh:EGPRs:RFTX:ENFTh` command.
Query	The query form of this command returns a string, containing `<int1>` floating point real numbers with the physical dimension of %, representing the measurement results. The single measurement results are separated by commas.

Example	`:MEAS:EGPRs:ARR:RFTX:ENFTh 5`
	`:FETC:EGPRs:RFTX:ENFTh?`
	The maximum error vector magnitude measurement will be performed five times. The measurement results will be stored internally and read by the `FETC` command. An example of the string returned is: "4.33,5.53,1.97,3.41,2.79".

:MEASure:EGPRs:ARRay:RFTX:EOFFset

Syntax	`:MEASure:EGPRs:ARRay:RFTX:EOFFset <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	Measures multiple times (set with the `int1` parameter) the Origin Offset. The origin offset indicates the accuracy of the i/Q modulators DC setup; it is usually an undesire leakage produced due to a DC offset in the phones I/Q modulator. It is measured in dBc. To read out the entire measurement results array, use the `:FETCh:EGPRs:RFTX:ENFTh` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of dBc, representing the measurement results. The single measurement results are separated by commas.

Example	`:MEAS:EGPRs:ARR:RFTX:EOFFset 5`
	`:FETC:EGPRs:RFTX:EOFFset?`
	The maximum error vector magnitude measurement will be performed five times. The measurement results will be stored internally and read by the `FETC` command. An example of the string returned is: "4.33,5.53,1.97,3.41,2.79".

:MEASure:EGPRs:ARRay:RFTX:FREQuency

Syntax	`:MEASure:EGPRs: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	Measures multiple times (set with the `int1` parameter) the frequency error. To read out the entire measurement results array, use the `:FETCh:EGPRs:RFTX:FREQ` command.
Query	The query form of this command returns a string, containing int1 floating point real numbers with the physical dimension of Hz, representing the measurement results. The single measurement results are separated by commas.

Example	`:MEAS:EGPRs:ARR:RFTX:FREQ 5`
	 `:FETC:EGPRs:RFTX:FREQ?`
	 The frequency error measurement will be performed five times. The measurement results will be stored internally and read by the `FETC` command. An example of the string returned is: `"22.33,22.53,21.97,22.41,21.79"`.

`:MEASure:EGPRs:ARRay:RFTX:LENGth`

Syntax	`:MEASure:EGPRs: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	Measures multiple times (set with the int1 parameter) the burst length. To read out the entire measurement results array, use the `:FETCh:EGPRs:RFTX:LENGth` command.
Query	The query form of this command returns a string, containing int1 floating point real numbers with the physical dimension of microseconds, representing the measurement results. The single measurement results are separated by commas.

Example	`:MEAS:EGPRs:ARR:RFTX:LENG 5`
	 `:FETC:EGPRs:RFTX:LENG?`
	 The measurement of the burst length will be performed five times. The measurement results will be stored internally and read by the `FETC` command. An example of the string returned is: `"556.4,557.2,558.2,557.8,556.5"`.

`:MEASure:EGPRs:ARRay:RFTX:UTIMe`

Syntax	`:MEASure:EGPRs: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	Measures multiple times (set with the int1 parameter) the uplink timing error. To read out the entire measurement results array, use the `:FETCh:EGPRs:RFTX:UTIMe` command.
Query	The query form of this command returns a string, containing `int1` floating point real numbers with the physical dimension of microseconds, representing the measurement results. The single measurement results are separated by commas.
Example	`:MEAS:EGPRs:ARR:RFTX:UTIM? 5` An example of the string returned is: `"0.0,0.1,0.0,-0.2,0.1"`.

:MEASure:EGPRs:ARRay:RFTX:POWer

Syntax	:MEASure:EGPRs:ARRay:RFTX:POWer <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	Measures multiple times (set with the int1 parameter) the burst peak power. To read out the entire measurement results array, use the :FETCh:EGPRs:RFTX:POWer command.
Query	The query form of this command returns a string, containing int1 floating point real numbers with the physical dimension of dBm, representing the measurement results. The single measurement results are separated by commas.
Example	:MEAS:EGPRs:ARR:RFTX:POW? 5 An example of the string returned is: "11.22,11.09,11.21,11.14,10.99".

:MEAS:EGPRs:ARR:RFTX:TEMP

Syntax	:MEAS:EGPRs:ARR:RFTX:TEMP <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	Checks multiple times (set with the int1 parameter) if TX burst fits into the power/time template. To read out the entire measurement results array, use the :FETCh:EGPRs:RFTX:TEMPlate command. **Note:** The power/time template is user-definable. It can be set up using the :CALC:EGPRs:RFTX:TEMP commands.
Query	The query form of this command returns a string containing int1 boolean numbers. A 0 indicates no violation of the PTT by the current burst, while a 1 signals PTT violation. The single measurement results are separated by commas.
Example	:MEAS:EGPRs:ARR:RFTX:TEMP 10 :FETC:EGPRs:RFTX:TEMP? The check of the current burst against the PTT will be performed ten times. The check resuls will be stored internally and read out by the FETC command. An example of the string returned is: "0,0,0,0,0,0,1,0,0,0", indicating that the seventh burst checked violated the PTT.

:MEASure:EGPRs:ARRay:RFTX:CORNer

Syntax	:MEASure:EGPRs: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	Measures multiple times (set with the int1 parameter) the RF power levels at eight user-definable positions specified by the CALC:EGPRs:RFTX:CORN:POS command. To read out the entire measurement results array, use the :FETCh:EGPRs:RFTX:CORNer command.

Query	The query form of this command returns a string, containing int1 times 8 floating point real numbers with the physical dimension of dBm, representing the measurement results. The single measurement results are separated by commas.

Example	`:MEAS:EGPRs:ARR:RFTX:CORN? 2`
	 An example of 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"`.

:MEAS:EGPRs:ARR:RFSP:ACPM:MODulation

Syntax	`:MEAS:EGPRs: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:EGPRs:RFSP:ACPM:MOD` command. **Note:** According to GSM specification 05.05, this measurement will be performed on ($\pm$)100 kHz and ($\pm$)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 ($\pm$)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:EGPRs:ARR:RFSPectrum:ACPM:MODulation 5`
	 `:FETCh:EGPRs:RFSP:ACPM:MOD?`
	 The string returned in this example would contain 115 floating point real numbers, representing the 23 measurement result values of five measurement runs.

:MEASure:EGPRs:ARRay:RFSPectrum:ACPM:MODPower

Syntax	`:MEASure:EGPRs: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:EGPRs: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:EGPRs:ARR:RFSPectrum:ACPM:MODPower 5`
	`:FETCh:EGPRs: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.

:MEASure:EGPRs:ARRay:RFSPectrum:ACPM:TRANsient

Syntax	`:MEASure:EGPRs:ARRay:RFSPectrum: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:EGPRs: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. 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 (27 * `int1`) floating point real numbers. The physical dimension is dBm. The single measurement results are separated by commas.

Example	`:MEAS:EGPRs:ARR:RFSPectrum:ACPM:TRANsient 5`
	`:FETCh:EGPRs:RFSP:ACPM:TRAN?`
	The string returned in this example would contain 135 floating point real numbers, representing the 27 measurement result values of five measurement runs.

:MEASure:EGPRs:ARRay:MULTislot:RFTX:TEMPlate

Syntax	`:MEASure:EGPRs:ARRay:MULTislot:RFTX:TEMPlate <int1>`
Parameters	`int1` specifies how many measurements shall be taken. Up to 100 measurements can be specified.
Description	Checks repetitively whether the `int1` TX bursts fit into the respective power/time templates.
Query	The return string contains int1 times the number of the uplink time slots followed by the pass/fail results for each of the individual slot. A 0 indicates that the burst shape is inside limits, 1 indicates outside limits.
Example	`CONF:EGPRs:MSTA:MSL 2`
	`:MEAS:EGPRs:ARR:MULT:RFTX:TEMP? 3`
	Returns a string like the following:
	`2,0,0 2,1,0, 2,0,0`
	In this example, the signal hit the limits in the first slot of the second measurement.

:MEASure:EGPRs:ARRay:MULTislot:RFSPectrum:ACPM:MODulation

Syntax	`:MEASure:EGPRs:ARRay:MULTislot:RFSPectrum:ACPM:MODulation <int1>`
Parameters	`int1` specifies how many measurements shall be taken. Up to 100 measurements are supported.
Description	Measures a defined number of times the RF output spectrum due to modulation on all the active uplink slots.
Query	Starts the measurement and returns int1 times the number of active uplink slots followed by 23 float values per time slot. For the meaning of the 23 float values, see `:MEAS:EGPRs:ACPM:MODulation`.
Example	`:CONF:EGPRs:MSTA:MSL 2`

:MEASure:EGPRs:ARRay:MULTislot:RFSPectrum:ACPM:TRANsient

Syntax	`:MEASure:EGPRs:ARRay:MULTislot:RFSPectrum:ACPM:`
	`:MEASure:EGPRs:ARRay:MULTislot:RFSPectrum:ACPM:`
	`TRANsient <int1>TRANsient <int1>`
Parameters	`int1` specifies how many measurements shall be taken. Up to 100 measurements are supported.
Description	Measures a defined number of times the RF output spectrum due to switching transients on all the active uplink slots.

Query	Starts the measurement and returns int1 times the number of active uplink slots followed by 27 float values per time slot. For the meaning of the 27 float values, see `:MEAS:EGPRs:ACPM:TRANsient`.
Example	`:CONF:EGPRs:MSTA:MSL 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: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 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: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: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 4400.

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 4400.

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 4400.

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 4400.

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 4400.

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 4400.

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 4400.

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 4400.
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 4400.

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 4400.

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 4400.

Example	`:MEAS:PSUP:PCUR`
	`:FETCh:PSUP:PCUR?`
	Returns a string like "**1256.2**" giving the peak current consumption in mA.

:FETCh:GPRS:RFTX:GROup?

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFTX:ALL` command.

Example	`:CONF:MEAS:GRO:RFTX POW,PRMS`
	`:MEAS:GPRS:RFTX:GRO`
	`:FETCh:GPRS: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:GPRS:RFTX:ALL?

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFTX:ALL`
	 `:FETCh:GPRS: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:GPRS:RFTX:PPEAk?`

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFTX:PPEAk`
	 `:FETCh:GPRS:RFTX:PPEAk?`
	 The value returned is: `"5.13"`.

`:FETCh:GPRS:RFTX:PRMS?`

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:ARR:RFTX:PRMS 5`
	`:FETCh:GPRS:RFTX:PRMS?`
	The values returned are: `"2.49,2.37,2.51,2.47,2.35"`.

`:FETCh:GPRS:RFTX:FREQuency?`

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:ARR:RFTX:FREQ 5`
	`:FETCh:GPRS:RFTX:FREQ?`
	The values returned are: `"22.49,24.37,21.51,29.47,23.35"`.

`:FETCh:GPRS:RFTX:LENGth?`

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFTX:LENGth`
	`:FETCh:GPRS:RFTX:LENG?`
	The value returned is: `"557.0"`.

`:FETCh:GPRS:RFTX:UTIMe?`

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:ARR:RFTX:UTIMe 5`
	`:FETCh:GPRS:RFTX:UTIM?`
	The values returned are: `"1.1,0.8,1.2,0.5,0.2"`.

:FETCh:GPRS:RFTX:POWer?

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFTX:POWer`
	`:FETCh:GPRS:RFTX:POWer?`
	The value returned is: `"13.05"`.

:FETCh:GPRS:RFTX:TEMPlate?

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFTX:TEMP` commands.

Example	`:MEAS:GPRS:ARR:RFTX:TEMPlate 10`
	`:FETCh:GPRS:RFTX:TEMP?`
	The values returned in this example are: `"0,0,0,0,0,0,0,0,0,0"`. This means that all the ten bursts checked were within the PTT.

:FETCh:GPRS:RFTX:CORNer?

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFTX:CORNer`
	`:FETCh:GPRS:RFTX:CORN?`
	The values returned are: `"-72.18,-61.91,-20.91,-0.05, -0.04,-17.97,-56.60,-73.95"`.

:FETCh:GPRS:RFTX:FLATness?

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFTX:FLATness`
	`:FETCh:GPRS:RFTX:FLAT?`
	The values returned are: `"-0.12,113.7,0.56,34.0"`.

:FETCh:GPRS:RFTX:FPOWer?

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFTX:FPOWer`
	`:FETCh:GPRS:RFTX:FPOWer?`
	The value returned is: `"13.05"`.

:FETCh:GPRS:BLOCkdata:BURStshape?

Syntax	`:FETCh:GPRS: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:GPRS:BLOC:BURS` 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:GPRS:BLOC:BURS`
	`:FETC:GPRS:BLOC:BURS?`
	The `FETC` command will return a data array, containing 711 floating point real numbers.

:FETCh:GPRS:BLOCkdata:PHASeerror?

Syntax	`:FETCh:GPRS: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:GPRS: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:GPRS:BLOC:PHAS`
	`:FETC:GPRS:BLOC:PHAS?`
	The `FETC` command will return a data array, containing 711 floating point real numbers.

:FETCh:GPRS:BLOCkdata:MSPectrum?

Syntax	`:FETCh:GPRS: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:GPRS:BLOC:MSP` 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:GPRS:BLOC:MSP`
	`:FETC:GPRS: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:GPRS:RFRX:RBER:ALL?

Syntax	`:FETCh:GPRS: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:GPRS[: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:GPRS:RFRX:RBER:ALL`
	`:FETCh:GPRS:RFRX:RBER:ALL?`
	This example will return the latest measurement result values for the three residual bit error measurements. The string delivered back in this example is:
	`"0.2,1.7,0.3"`. This means that the RBER for class Ib bits is `0.2`.

:FETCh:GPRS:RFRX:RBER:CIB?

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFRX:RBER:CIB` measurement. The string delivered back will contain at least one floating point real number.

Example	`:MEAS:GPRS:ARR:RFRX:RBER:CIB 5`
	`:FETCh:GPRS:RFRX:RBER:CIB?`
	The string delivered back is:
	`"0.2,0.1,0.2,0.3,0.2"`.

:FETCh:GPRS:RFRX:RBER:CII?

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFRX:RBER:CII` measurement. The string delivered back will contain at least one floating point real number.

Example	`:MEAS:GPRS:ARR:RFRX:RBER:CII 5`
	`:FETCh:GPRS:RFRX:RBER:CII?`
	The string delivered back is:
	`"1.5,1.7,2.1,2.3,2.6"`.

:FETCh:GPRS:RFRX:RBER:FER?

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFRX:RBER:FER` measurement. The string delivered back will contain at least one floating point real number.

Example	`:MEAS:GPRS:RFRX:RBER:FER`
	`:FETCh:GPRS:RFRX:RBER:FER?`
	The string delivered back is:
	`"1.5"`.

`:FETCh:GPRS:RFRX:BER:ALL?`

Syntax	`:FETCh:GPRS: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:GPRS[: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 la bits, 2. the result of the (nonresidual) bit error ratio measurement of the class lb bits, and 3. the result of the (nonresidual) bit error ratio measurement of the class II bits.

Example	`:MEAS:GPRS:ARR:RFRX:BER:ALL 3`
	`:FETCh:GPRS:RFRX:BER:ALL?`
	This example will deliver back the latest measurement result values for the three residual bit error measurements. The string delivered back in this example is:
	`"0.2,0.3,1.7,0.2,0.4,1.8,0.1,0.2,0.8"`.

`:FETCh:GPRS:RFRX:BER:CIA?`

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFRX:BER:CIA` measurement. The string delivered back will contain at least one floating point real number.

Example	`:MEAS:GPRS:RFRX:BER:CIA`
	`:FETCh:GPRS:RFRX:BER:CIA?`
	The string delivered back is:
	`"0.1"`.

`:FETCh:GPRS:RFRX:BER:CIB?`

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFRX:BER:CIB` measurement. The string delivered back will contain at least one floating point real number.

Example	`:MEAS:GPRS:ARR:RFRX:BER:CIB 5`
	`:FETCh:GPRS:RFRX:BER:CIB?`
	The string delivered back is:
	`"0.4,0.3,0.7,0.6,0.5"`.

:FETCh:GPRS:RFRX:BER:CII?

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFRX:BER:CII` measurement. The string delivered back will contain at least one floating point real number.

Example	`:MEAS:GPRS:RFRX:BER:CII`
	`:FETCh:GPRS:RFRX:BER:CII?`
	The string delivered back is:
	`"1.9"`.

:FETCh:GPRS:RFRX:BER:CRC?

Syntax	`:FETCh:GPRS:RFRX:BER:CRC?`
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:GPRS[:ARRay]:RFRX:BER:CII` measurement. The string delivered back will contain at least one floating point real number.

Example	`:MEAS:GPRS:RFRX:BER:CRC`
	`:FETC:GPRS:RFRX:BER:CRC?`
	The string delivered back in this example is:
	`"1.9"`.

:FETCh:GPRS:RFRX:FBER?

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFRX:FBER` measurement. The string delivered back will contain at least one floating point real number.

Example	`:MEAS:GPRS:ARR:RFRX:FBER 5`
	`:FETCh:GPRS:RFRX:FBER?`
	The string delivered back is:
	`"0.7,0.9,1.2,1.8,2.0"`.

:FETCh:GPRS:RFRX:RTDelay?

Syntax	`:FETCh:GPRS: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 `:MEAS:GPRS[:ARRay]:RFRX:RTDelay` measurement. The string delivered back will contain at least one integer.

Example	`:MEAS:GPRS:RFRX:RTD`
	`:FETCh:GPRS:RFRX:RTD?`
	The string delivered back is:
	`"8"`.

:FETCh:GPRS:RFSP:ACPM:MODulation?

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFSPectrum:ACPM:MOD` command.

Example	`:MEAS:GPRS:RFSPectrum:ACPM:MOD`
	`:FETC:GPRS:RFSP:ACPM:MOD?`
	The `FETC` command will return a data array, containing 23 floating point real numbers.

:FETCh:GPRS:RFSPectrum:ACPM:MODPower?

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFSPectrum:ACPM:MOD` command.

Example	`:MEAS:GPRS:RFSP:ACPM:MOD`
	`:FETC:GPRS:RFSP:ACPM:MOD?`
	The `FETC` command will return a data array, containing 24 floating point real numbers (the power reference value and 23 values as in `ACPM:MOD`).

:FETCh:GPRS:RFSP:ACPM:TRANsient?

Syntax	`:FETCh:GPRS: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:GPRS[:ARRay]:RFSP:ACPM:TRAN` command.

Example	`:MEAS:GPRS:RFSPectrum:ACPM:TRAN`
	`:FETC:GPRS:RFSP:ACPM:TRAN?`
	The `FETC` command will return a data array, containing 27 floating point real numbers.

:FETCh:GPRS:MULTislot:RFTX:ALL?

Syntax	`:FETCh:GPRS:MULTislot:RFTX:ALL?`
Parameters	There are no parameters with this command.
Description	Only the query form of this command is available.
Query	Requires a preceding `:MEAS:GPRS:ARR:MULT:RFTX:ALL <int1>` command. Returns a string containing `int1` times the result string defined for the `:FETCh:GPRS:MULT:RFTX:ALL` query.

Example	`:CONF:GPRS:MSTA:MSL 2`
	`:MEAS:GPRS:ARR:MULT:RFTX:ALL 10`
	`:FETC:GPRS:MULT:RFTX:ALL?`
	Delivers a string containing 10*(1+2*19)=390 results, separated by commas.

:FETCh:GPRS:MULTIslot:RFTX:PPEAk?

Syntax	`:FETCh:GPRS:MULTIslot:RFTX:PPEAk?`
Parameters	There are no parameters with this command.
Description	Only the query form of this command is available.
Query	Returns a string containing `int1` times the result string from the `:MEASure:GPRS:CONT:MULT:RFTX:PPEAk` query. `int1` is specified with the preceding `:MEAS:GPRS:ARR:MULT:RFTX:PPEAk` command.

Example	`:CONF:GPRS:MSTA:MSL 2`
	`:MEAS:GPRS:ARR:MULT:RFTX:PPEAk 3`
	`:FETCh:GPRS:MULT:RFTX:PPEAk?`
	Returns a string containing nine results separated by commas, for example
	`2,8.3,9.6, 2,8.1,8.9, 2,8.2,9.5`

:FETCh:GPRS:MULTIslot:RFTX:TEMPlate?

Syntax	`:FETCh:GPRS:MULTIslot:RFTX:TEMPlate?`
Parameters	There are no parameters with this command.
Description	Only the query form of this command is available.
Query	The return string contains `int1` times the number of the uplink time slots followed by the pass/fail results for the individual slots, one result for each slot. A 0 indicates that the burst shape is inside limits, 1 indicates outside limits for the respective part. `int1` is specified with the preceding `:MEAS:GPRS:ARR:MULT:RFTX:TEMP` command.

Example	`:CONF:GPRS:MSTA:MSL 2`
	`:MEAS:GPRS:ARR:MULT:RFTX:TEMP 3`
	`:FETCh:GPRS:MULT:RFTX:TEMP?`
	Returns a string like the following:
	`2,0,0, 2,0,1, 2,0,1`
	In this example, the signal hits the limits in the second slot for the second and third measurement.

:FETCh:GPRS:MULTIslot:RFSPectrum:ACPM:MODulation?

Syntax	`:FETCh:GPRS:MULTIslot:RFSPectrum:ACPM:MODulation?`
Parameters	There are no parameters with this command.
Description	Only the query form of this command is available.
Query	Returns `int1` times the number of active uplink slots followed by 23 float values per time slot. `int1` is the number of measurements defined in the respective MEAS:GPRS:ARRay command. For the meaning of the 23 float values per slot, see `:MEAS:GPRS:ACPM:MODulation`.

Example	`:CONF:GPRS:MSTA:MSL 2`
	`:MEAS:GPRS:ARR:MULT:RFSP:ACPM:MOD 3`
	`:FETCh:GPRS:MULT:RFSP:ACPM:MOD?`

:FETCh:GPRS:MULTIslot:RFSPectrum:ACPM:TRANsient?

Syntax	`:FETCh:GPRS:MULTIslot:RFSPectrum:ACPM:TRANsient?`
Parameters	There are no parameters with this command.
Description	Only the query form of this command is available.
Query	Returns `int1` times the number of active uplink slots followed by 27 float values per time slot. For the meaning of the 27 float values, see `:MEAS:GPRS:ACPM:TRANsient`.

Example	`:CONF:GPRS:MSTA:MSL 2`
	`:FETCh:GPRS:MULT:RFSP:ACPM:TRAN?`

`:FETCh:GPRS:MULTIslot:RFRX:BLER?`

Syntax	`:FETCh:GPRS:MULTIslot:RFRX:BLER?`
Parameters	There are no parameters with this command.
Description	Only the query form of this command is available.
Query	Returns a string containing `int1` times the number of active slots followed by the block error rate (BLER) on each of these time slots. The BLER is expressed in % and a floating point real value.

Example	`:CONF:GPRS:MSTA:MSL 2`
	`:CALL:GPRS:MODE USF`
	`:MEAS:GPRS:ARR:MULT:RFRX:BLER 3`
	`:FETCh:GPRS:MULT:RFRX:BLER?`
	Returns a string like
	`2,0.7,1.4, 2,0.9,1.5, 2,0.8,1.3`

`:FETCh:EGPRs:RFTX:GROup?`

Syntax	`:FETCh:EGPRs: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 EGPRS RF TX measurements as specified with the `:CONF:EGPRs:MEAS:GRO:RFTX` command and measured with the `:MEAS:EGPRs[:ARRay]:RFTX:GROup` 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 `:FETCh:EGPRs:RFTX:ALL` command.

Example	`:CONF:EGPR:MEAS:GRO:RFTX POW,ERMS`
	`:MEAS:EGPR:RFTX:GRO`
	`:FETC:EGPR:RFTX:GRO?`
	In this example, the group of measurements is defined by a power level measurement combined with a EVM RMS phase error measurement. The string returned in this example 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 EVM RMS phase error, the second one the mobile's RF output power level.

:FETCh:EGPRs:RFTX:ALL?

Syntax	:FETCh:EGPRs: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:EGPR[:ARR]:RFTX:ALL measurement. The string delivered back will contain 17 measurement result values, separated with commas. The order and type of these measurement result values delivered back is as follows: 1. ERMS, the error vector magnitude rms-averaged over a burst. EVM is a relative value and shown as a percentage, 2. EPEAk, the maximum error vector magnitude within a burst. EVM is indicated per cent, 3. ENFTh, the error vector magnitude that is not exceeded by 95% of the symbols of a burst. The 95% EVM is shown in %, 4. EOFFset, the origin offset indicates the accuracy of the i/Q modulators DC setup; it is usually an undesire leakage produced due to a DC offset in the phones I/Q modulator. It is measured in dBc, 5. FREQuency, floating point real number representing the mobile's frequency error, 6. LENGth, floating point real number representing the length of the burst in microseconds, 7. UTIMe, floating point real number representing the timing error of the mobile under test in microseconds (taking the ordered timing advance into account), 8. POWer, floating point real number representing the mobile's RF output power level in dBm, 9. TEMPlate boolean number. 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. 10...17. CORNer, eight floating point real numbers, representing the RF power level in dBm at eight pre-defined positions of a burst.
Example	:MEAS:EGPR:RFTX:ALL :FETC:EGPR:RFTX:ALL? This example will return the latest measurement result values for the 17 most important EGPRS RF TX measurements. The string delivered back in this example is: "5.13,11.94,3,12,4.54,-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".

:FETCh:EGPRs:RFTX:ERMS?

Syntax	:FETCh:EGPRs:RFTX:ERMS?
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:EGPR[:ARR]:RFTX:ERMS measurement. The string delivered back will contain at least one floating point real number with the physical dimension of %.

Example	`:MEAS:EGPR:ARR:RFTX:ERMS 5`
	`:FETC:EGPR:RFTX:ERMS?`
	The error vector magnitude rms-averaged measurement will be performed five times. The measurement results will be stored internally and read by the FETC command. An example of the string returned is: `"2.33,2.53,1.97,2.41,1.79"`.

`:FETCh:EGPRs:RFTX:EPEAk?`

Syntax	`:FETCh:EGPRs:RFTX:EPEAk?`
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:EGPR[:ARR]:RFTX:EPEA` measurement. The string delivered back will contain at least one floating point real number with the physical dimension of %.

Example	`:MEAS:EGPR:ARR:RFTX:EPEA 5`
	`:FETC:EGPR:RFTX:EPEA?`
	The maximum error vector magnitude measurement will be performed five times. The measurement results will be stored internally and read by the FETC command. An example of the string returned is: `"6.33,8.53,11.97,9.41,10.79"`.

`:FETCh:EGPRs:RFTX:ENFTh?`

Syntax	`:FETCh:EGPRs:RFTX:ENFTh?`
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:EGPR[:ARR]:RFTX:ENFT` measurement. The string delivered back will contain at least one floating point real number with the physical dimension of %.

Example	`:MEAS:EGPR:ARR:RFTX:ENFT 5`
	`:FETC:EGPR:RFTX:ENFT?`
	The maximum error vector magnitude measurement will be performed five times. The measurement results will be stored internally and read by the FETC command. An example of the string returned is: `"4.33,5.53,1.97,3.41,2.79"`.

`:FETCh:EGPRs:RFTX:EOFFset?`

Syntax	`:FETCh:EGPRs:RFTX:EOFFset?`
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:EGPR[:ARR]:RFTX:EOFF` measurement. The string delivered back will contain at least one floating point real number with the physical dimension of dBc.

Example	`:MEAS:EGPR:ARR:RFTX:EOFF 5`
	`:FETC:EGPR:RFTX:EOFF?`
	The maximum error vector magnitude measurement will be performed five times. The measurement results will be stored internally and read by the FETC command. An example of the string returned is: `"4.33,5.53,1.97,3.41,2.79"`.

`:FETCh:EGPRs:RFTX:FREQuency?`

Syntax	`:FETCh:EGPRs: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:EGPRs[: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:EGPRs:ARR:RFTX:FREQ 5`
	`:FETCh:EGPRs:RFTX:FREQ?`
	The values returned in this example are:
	`"22.49,24.37,21.51,29.47,23.35"`.

`:FETCh:EGPRs:RFTX:LENGth?`

Syntax	`:FETCh:EGPRs: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:EGPRs[: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:EGPRs:RFTX:LENGth`
	`:FETCh:EGPRs:RFTX:LENG?`
	The value returned in this example is: `"557.0"`.

`:FETCh:EGPRs:RFTX:UTIMe?`

Syntax	`:FETCh:EGPRs: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:EGPRs[: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:EGPRs:ARR:RFTX:UTIMe 5`
	`:FETCh:EGPRs:RFTX:UTIM?`
	The values returned in this example are: "`1.1,0.8,1.2,0.5,0.2`".

`:FETCh:EGPRs:RFTX:POWer?`

Syntax	`:FETCh:EGPRs: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:EGPRs[: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:EGPRs:RFTX:POWer`
	`:FETCh:EGPRs:RFTX:POWer?`
	The value returned in this example is: "`13.05`".

`:FETCh:EGPRs:RFTX:TEMPlate?`

Syntax	`:FETCh:EGPRs: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:EGPRs[: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:EGPRs:RFTX:TEMP` commands.
Example	`:MEAS:EGPRs:ARR:RFTX:TEMPlate 10`
	`:FETCh:EGPRs:RFTX:TEMP?`
	The values returned in this example are: "`0,0,0,0,0,0,0,0,0,0`". This means that all the ten bursts checked were within the PTT.

`:FETCh:EGPRs:RFTX:CORNer?`

Syntax	`:FETCh:EGPRs: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:EGPRs[: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:EGPRs:RFTX:CORNer`
	`:FETCh:EGPRs:RFTX:CORN?`
	The values returned in this example are: "`-72.18,-61.91,-20.91,-0.05,`
	`-0.04,-17.97,-56.60,-73.95`".

`:FETCh:EGPRs:BLOCkdata:BURStshape?`

Syntax	`:FETCh:EGPRs: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:EGPRs:BLOC:BURS` 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:EGPRs:BLOC:BURS`
	`:FETC:EGPRs:BLOC:BURS?`
	The `FETC` command will return a data array, containing 711 floating point real numbers.

`:FETCh:EGPRs:BLOCkdata:MSPectrum?`

Syntax	`:FETCh:EGPRs: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:EGPRs:BLOC:MSP` 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:EGPRs:BLOC:MSP`
	`:FETC:EGPRs: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:EGPRs:RFSP:ACPM:MODulation?`

Syntax	`:FETCh:EGPRs: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:EGPRs[:ARRay]` `:RFSPectrum:ACPM:MOD` command.

Example	`:MEAS:EGPRs:RFSPectrum:ACPM:MOD`
	`:FETC:EGPRs:RFSP:ACPM:MOD?`
	The `FETC` command will return a data array, containing 23 floating point real numbers.

:FETCh:EGPRs:RFSP:ACPM:MODPower?

Syntax	`:FETCh:EGPRs:RFSP: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:EGPRs[:ARRay]` `:RFSPectrum:ACPM:MOD` command.

Example	`:MEAS:EGPRs:RFSPectrum:ACPM:MOD`
	`:FETC:EGPRs:RFSP:ACPM:MOD?`
	The `FETC` command will return a data array, containing 24 floating point real numbers (the power reference value and 23 values as in `ACPM:MOD`).

:FETCh:EGPRs:RFSP:ACPM:TRANsient?

Syntax	`:FETCh:EGPRs: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:EGPRs[:ARRay]:RFSP:ACPM:TRAN` command.

Example	`:MEAS:EGPRs:RFSPectrum:ACPM:TRAN`
	`:FETC:EGPRs:RFSP:ACPM:TRAN?`
	The `FETC` command will return a data array, containing 27 floating point real numbers.

:FETCh:EGPRs:MULTIslot:RFTX:TEMPlate?

Syntax	`:FETCh:EGPRs:MULTIslot:RFTX:TEMPlate?`
Parameters	There are no parameters with this command.
Description	Only the query form of this command is available.

4463 GSM/GPRS System Option and 4468 EDGE System Option   Version 6.20

Query	The return string contains `int1` times the number of the uplink time slots followed by the pass/fail results for the individual slots, one result for each slot. A `0` indicates that the burst shape is inside limits, `1` indicates outside limits for the respective part. `int1` is specified with the preceding `:MEAS:EGPRs:ARR:MULT:RFTX:TEMP` command.

Example	`:CONF:EGPRs:MSTA:MSL 2`
	`:MEAS:EGPRs:ARR:MULT:RFTX:TEMP 3`
	`:FETCh:EGPRs:MULT:RFTX:TEMP?`
	Returns a string like the following:
	`2,0,0, 2,0,1, 2,0,1`
	In this example, the signal hit the limits for the second slot in the second and third measurement.

`:FETCh:EGPRs:MULTIslot:RFSPectrum:ACPM:MODulation?`

Syntax	`:FETCh:EGPRs:MULTIslot:RFSPectrum:ACPM:MODulation?`
Parameters	There are no parameters with this command.
Description	Only the query form of this command is available.
Query	Returns `int1` times the number of active uplink slots followed by 23 float values per time slot. `int1` is the number of measurements defined in the respective `MEAS:EGPRs:ARRay` command. For the meaning of the 23 float values per slot, see `:MEAS:EGPRs:ACPM:MODulation`.

Example	`:CONF:EGPRs:MSTA:MSL 2`
	`:MEAS:EGPRs:ARR:MULT:RFSP:ACPM:MOD 3`
	`:FETCh:EGPRs:MULT:RFSP:ACPM:MOD?`

`:FETCh:EGPRs:MULTIslot:RFSPectrum:ACPM:TRANsient?`

Syntax	`:FETCh:EGPRs:MULTIslot:RFSPectrum:ACPM:TRANsient?`
Parameters	There are no parameters with this command.
Description	Only the query form of this command is available.
Query	Returns `int1` times the number of active uplink slots followed by 27 float values per time slot. For the meaning of the 27 float values, see `:MEAS:EGPRs:ACPM:TRANsient`.
Example	`:CONF:EGPRs:MSTA:MSL 2` `:FETCh:EGPRs:MULT:RFSP:ACPM:TRAN?`

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. For further details, refer to the example shown in section "The CALCulate Subsystem" on page 375.
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:RESet

Syntax	:CALCulate:LIMit:FAIL: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: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:GSM:RFTX:FREQ`
	`:CALC:RES`
	`...(other SCPI commands)...`
	`:CALC:MSIG?`
	In this example, first a continuous measurement of the mobile's frequency error is started. After some time, the `:CALC:MSIG` command is used to read out the mean frequency error and its standard deviation. The string returned in this example is `"22.1, 3.4"`

`: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 4400.
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 4400.
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.

<div align="center">

:CALCulate:AFSPectrum:TEMPlate:LIMit[:FAIL]

</div>

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.

<div align="center">

:CALCulate:AFSPectrum:TEMPlate:LIMit:STATe

</div>

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.

<div align="center">

:CALCulate:AFSPectrum:TEMPlate:LIMit:LOAD

</div>

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 4400.
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 4400.
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 4400.
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:GPRS:RFTX:MAVerage?

Syntax	`:CALCulate:GPRS: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:MEAS:GRO:RFTX POW,PRMS`
	`:CALC:RES`
	`:MEAS:GPRS:ARR:RFTX:GRO 20`
	`:CALC:GPRS: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 `:MEASure:GPRS[:ARRay]:RFTX:ALL` command), the first measurement result value delivered back is the average RMS phase error, the second one the mobile's average RF output power.

:CALCulate:GPRS:RFTX:MMINimum?

Syntax	`:CALCulate:GPRS: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:MEAS:GRO:RFTX POW,PRMS`
	`:CALC:RES`
	`:MEAS:GPRS:ARR:RFTX:GROup 20`
	`:CALCulate:GPRS: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 `:MEASure:GSM[:ARRay]: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:GPRS:RFTX:MMAXimum?

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:FREQ`
	`:CALC:RES`
	`...(other SCPI commands)...`
	`:CALC:GPRS:RFTX: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 deviation of the mobile. The string returned in this example is `"22.1"`.

:CALCulate:GPRS:RFTX:MSIGma?

Syntax	`:CALCulate:GPRS: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 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 are separated by commas.

Example	`:MEAS:GPRS:RFTX:FREQ`
	`:CALC:RES`
	`...(other SCPI commands)...`
	`:CALC:GPRS:RFTX:MSIG?`
	In this example, first a continuous measurement of the mobile's frequency error is started. After some time, the `:CALC:GPRS: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:GPRS:RFTX:ALL:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS: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:GPRS[: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.

Example	`:CALC:GPRS:RFTX:ALL:LIM?`
	String returned:
	`"0,0,0,0,0,1,0,0,0,0,0,0,0,0,0"`
	The 1 on position 6 indicates that the mobile's RF power level violated its test limits.

:CALCulate:GPRS:RFTX:ALL:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:ALL` command for reference).
Query	There is no query form of this command available.
Example	`:CALC:GPRS:RFTX:ALL:LIM:STAT OFF` Switches the limit check for the most relevant RF TX measurements off.

:CALCulate:GPRS
:RFTX:ALL:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GPRS :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:GPRS: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:GPRS:RFTX:ALL:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS :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:GPRS: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.

:CALC:GPRS:RFTX:PPEAk:LIMit[:FAIL]?

Syntax	:CALC:GPRS: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:GPRS:ARR:RFTX:PPEA 20
	:CALC:GPRS:RFTX:PPEA: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:GPRS:RFTX:PPEAk:LIMit:STATe

Syntax	:CALCulate:GPRS: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:GPRS:RFTX:PPEA measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	:CALC:GPRS:RFTX:PPEAk:LIM:STAT OFF
	Switches the limit check for the peak phase error measurement off.

:CALCulate:GPRS:RFTX:PPEAk:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GPRS: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:GPRS:RFTX:PPEA:LIM:UPP 15
	Sets the upper limit for the peak phase error to 15 degrees.

:CALCulate:GPRS:RFTX:PPEAk:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS: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:GPRS:RFTX:PPEA:LIM:LOW −15
	Sets the lower limit for the peak phase error measurement to −15 degrees.

:CALCulate:GPRS:RFTX:PRMS:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS: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:GPRS:ARR:RFTX:PRMS 10`
	`:CALC:GPRS:RFTX:PRMS:LIM?`
	String returned:`"0"`
	This string delivered back indicates that there was no violation of the measurement limits.

:CALCulate:GPRS:RFTX:PRMS:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:PRMS` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	`:CALC:GPRS:RFTX:PRMS:LIM:STAT OFF` Switches the limit check for the RMS-valued phase error measurement off.

:CALCulate:GPRS:RFTX:PRMS:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:PRMS:LIM:UPP 6.2` Sets the upper limit for the RMS-valued phase error measurement to 6.2 degrees.

:CALCulate:GPRS:RFTX:PRMS:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:PRMS:LIM:LOW 0.0` Sets the lower limit for the RMS-valued phase error measurement to 0 degrees.

:CALCulate:GPRS:RFTX:FREQuency:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS: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:GPRS:ARRay:RFTX:FREQ 20`
	`:CALC:GPRS:RFTX:FREQ:LIM?`
	String returned:`"0"`
	This string delivered back indicates that there was no violation of the measurement limits.

:CALCulate:GPRS:RFTX:FREQuency:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:FREQ` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	`:CALC:GPRS:RFTX:FREQ:LIM:STAT ON` Switches the limit check for the frequency error measurement on.

:CALCulate:GPRS:RFTX:FREQuency:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS: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	There is no query form of this command available.
Query	The query form of this command delivers the current setting of the command. The string returned will contain one floating point real number.
Example	`:CALC:GPRS:RFTX:FREQ:LIM:UPP 45` Sets the upper limit of the frequency error measurement result to 45 Hz.

:CALCulate:GPRS:RFTX:FREQuency:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS: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 measurement; the physical dimension of the number stated is Hertz.
Query	There is no query form of this command available.
Example	:CALC:GPRS:RFTX:FREQ:LIM:LOW −60.8 Sets the lower limit of the frequency error measurement result to −60.8 Hz.

:CALCulate:GPRS:RFTX:FREQuency:LIMit:GSM

Syntax	:CALCulate:GPRS: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:GPRS: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:GPRS:RFTX:FREQuency:LIMit:PCN

Syntax	:CALCulate:GPRS: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:GPRS: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:GPRS:RFTX:FREQuency:LIMit:PCS

Syntax	:CALCulate:GPRS: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:GPRS: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:GPRS:RFTX:LENGth:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS: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:GPRS:ARR:RFTX:LENG 10`
	 `:CALC:GPRS:RFTX:LENG:LIM?`
	 String returned:`"0"`
	 This string indicates that there was no violation of the measurement limits.

:CALCulate:GPRS:RFTX:LENGth:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:LENGth` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	`:CALC:GPRS:RFTX:LENG:LIM:STAT ON` Switches the limit check for the burst length measurement on.

:CALCulate:GPRS:RFTX:LENGth:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS:RFTX:LENGth:LIMit:UPPer[: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:GPRS:RFTX:LENG:LIM:UPP 563 Sets the upper limit of the burst length to 563.0 µs.

:CALCulate:GPRS:RFTX:LENGth:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS: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:GPRS:RFTX:LENG:LIM:LOW 541.9 Sets the lower limit of the burst length to 541.9 µs.

:CALCulate:GPRS:RFTX:UTIMe:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS: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:GPRS:ARR:RFTX:UTIM 10 :CALC:GPRS:RFTX:UTIM:LIM? String returned:"0" This string indicates that there was no violation of the measurement limits.

:CALCulate:GPRS:RFTX:UTIMe:LIMit:STATe

Syntax	:CALCulate:GPRS: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:GPRS:RFTX:UTIMe measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	:CALC:GPRS:RFTX:UTIM:LIM:STAT ON Switches the limit check for the UL timing error measurement on.

`:CALCulate:GPRS:RFTX:UTIMe:LIMit:UPPer[:DATA]`

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:UTIM:LIM:UPP 5` Sets the upper limit of the uplink timing error measurement result to 5.0 µs.

`:CALCulate:GPRS:RFTX:UTIMe:LIMit:LOWer[:DATA]`

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:UTIM:LIM:LOW -5` Sets the lower limit of the uplink timing error measurement result to –5.0 µs.

`:CALCulate:GPRS:RFTX:POWer:LIMit[:FAIL]?`

Syntax	`:CALCulate:GPRS: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:GPRS:ARRay:POW 20`
	`:CALC:GPRS:RFTX:POW:LIM?`
	String returned:`"0"`
	This string indicates that there was no violation of the measurement limits.

:CALCulate:GPRS:RFTX:POWer:LIMit:STATe

Syntax	:CALCulate:GPRS: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:GPRS:RFTX:POWer measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	:CALC:GPRS:RFTX:POW:LIM:STAT ON Switches the limit check for the RF output power level measurement on.

:CALCulate:GPRS:RFTX:POWer:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GPRS: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:GPRS:RFTX:POW:LIM:UPP 13 Sets the upper limit of the peak power measurement to 13 dBm.

:CALCulate:GPRS:RFTX:POWer:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS: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:GPRS:RFTX:POW:LIM:LOW -45 Sets the lower limit of the peak power measurement to -45 dBm.

:CALCulate:GPRS:RFTX:POWer:LIMit:GSM

Syntax	:CALCulate:GPRS: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, 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:GPRS:RFTX:POW:LIM:GSM 2, 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, 5, 5, 5, 5 dB around the nominal level.

:CALCulate:GPRS:RFTX:POWer:LIMit:PCN

Syntax	`:CALCulate:GPRS: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:GPRS: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, 4, 4, 4, 4, 5, 5 dB around the nominal level.

:CALCulate:GPRS:RFTX:POWer:LIMit:PCS

Syntax	`:CALCulate:GPRS: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, 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:GPRS: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:GPRS:RFTX:TEMPlate[:NORMal]:LIMit [:FAIL]?

Syntax	`:CALCulate:GPRS: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:GPRS: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:GPRS:RFTX:TEMP`
	`:CALC:GPRS: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:GPRS:RFTX:TEMPlate[:NORMal]:LIMit:AUTO

Syntax	`:CALCulate:GPRS:RFTX:TEMPlate[:NORMal]:LIMit:AUTO <Pre-defExp>`
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 4400 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 4400 will use the manually set limits instead (see description of the `:CALC:GPRS:RFTX:TEMP:LIM:UPP` and `:CALC:GPRS:RFTX:TEMP:LIM:LOW` commands for further details).
Query	There is no query form of this command available.
Example	`:CALC:GPRS:RFTX:TEMP:LIM:AUTO ON` Switches the limit check for the burst shape to use the standard PTTs as defined by ETSI standards.

:CALCulate:GPRS:RFTX:TEMPlate[:NORMal]:LIMit:STAT

Syntax	:CALCulate:GPRS:RFTX:TEMPlate[:NORMal]:LIMit:STAT <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:GPRS:RFTX:TEMPlate measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	:CALC:GPRS:RFTX:TEMP:LIM:STAT ON Switches on the limit check of the burst shape against the PTT for normal bursts.

**:CALCulate:GPRS
:RFTX:TEMPlate[:NORMal]:LIMit:UPPer[:DATA]**

Syntax	:CALCulate:GPRS :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:GPRS:RFTX:TEMP:LIM:UPP -41,-47,-28,-47,-18,-28,-` `10,-4,0 ,4,553,1,561,-4,571,-28,580,-47`

:CALCulate:GPRS
:RFTX:TEMPlate[:NORMal]:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:GPRS`
	`: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:GPRS:RFTX:TEMP:LIM:LOW` `-41,-150,0,-150,0,-40,20, -1,270,-1,543,-1,543,-150`

:CALCulate:GPRS:RFTX:TEMPlate:RACH:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS` `: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:GPRS: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:GPRS: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:GPRS:RFTX:TEMPlate:RACH:LIMit:AUTO

Syntax	`:CALCulate:GPRS: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 4400 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 4400 will use the manually set limits instead (see description of the `:CALC:GPRS:RFTX:TEMP:RACH:LIM:UPP` and `:CALC:GPRS:RFTX:TEMP:RACH:LIM:LOW` commands for further details).
Query	There is on query form to this commmand.
Example	`:CALC:GPRS: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:GPRS:RFTX:TEMPlate:RACH:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS:RFTX:TEMP:RACH:LIM:STAT ON` Switches on the limit check of the shape of random access bursts against the PTT.

:CALCulate:GPRS
:RFTX:TEMPlate:RACH:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS`
	`: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 available.

Example	`:CALC:GPRS:RFTX:TEMP:RACH:LIM:UPP`
	`-41,-59,-28,-59,-18,-30,-10,-6,0,  4,331,1,339,-6,349,`
	`-30,349,-59`

:CALCulate:GPRS
:RFTX:TEMPlate:RACH:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS :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 available.
Example	:CALC:GSM:RFTX:TEMP:RACH:LIM:LOW –41,–150,0,–150,0,–40,20, –1,270,–1,321,–1,321,–150

:CALCulate:GPRS
:RFTX:CORNer[:NORMal]:POSition

Syntax	:CALCulate:GPRS :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:GPRS:RFTX:CORN:POS -28,-18,- 10,0,542.8,552.8,562.8,572.8

:CALCulate:GPRS
:RFTX:CORNer[:NORMal]:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS :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:GPRS:RFTX:CORN:NORM:POS command as described above. Please refer to section Corner Points for further details.
Example	:MEAS:GPRS:ARRay:CORN 10 :CALC:GPRS:RFTX:CORN:LIM:FAIL? String returned:"0,0,1,0,0,0,0,0" This string indicates that the limits set for corner point 3 were violated.

:CALCulate:GPRS
:RFTX:CORNer[:NORMal]:LIMit:STATe

Syntax	:CALCulate:GPRS :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:GPRS:RFTX:CORNer measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	:CALC:GPRS:RFTX:CORN:LIM:STAT OFF Switches the limit check at the eight corner positions off.

:CALCulate:GPRS
:RFTX:CORNer[:NORMal]:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GPRS :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. For further details, please refer to section Corner Points.
Query	There is no query form to this command.
Example	:CALC:GPRS:RFTX:CORN:LIM:UPP 4,4,4,4,4,4,4,4

:CALCulate:GPRS
:RFTX:CORNer[:NORMal]:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS :RFTX:CORNer[:NORMal]: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. For further details, please refer to section Corner Points.
Query	There is no query form to this command.

Example	:CALC:GPRS: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

:CALCulate:GPRS:RFRX:MAVerage?

Syntax	:CALCulate:GPRS: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:GPRS:ARR:RFRX:RBER:ALL :CALC:GPRS:RFRX:MAV?

:CALCulate:GPRS:RFRX:MMINimum?

Syntax	:CALCulate:GPRS: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:GPRS:ARR:RFRX:RBER:ALL`
	`:CALC:GPRS:RFRX:MMIN?`

:CALCulate:GPRS:RFRX:MMAXimum?

Syntax	`:CALCulate:GPRS: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:GPRS:ARR:RFRX:RBER:ALL`
	`:CALC:GPRS:RFRX:MMAX?`

:CALCulate:GPRS:RFRX:MSIGma?

Syntax	`:CALCulate:GPRS: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:GPRS:RFRX:RBER:CII`
	`...(other SCPI commands)...`
	`:CALC:GPRS: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:GPRS: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:GPRS:RFRX:RBER:ALL:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS` `: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:GPRS:RFRX:RBER:ALL` command.

Example	`:MEAS:GPRS:RFRX:RBER:ALL`
	`:CALC:GPRS:RFRX:RBER:ALL:LIM?v`
	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:GPRS:RFRX:RBER:ALL:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS: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:GPRS:RFRX:RBER:ALL:LIM:STAT OFF` Switches the limit check of all residual bit error ratio measurements off.

:CALCulate:GPRS:RFRX:RBER:ALL:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS`
	`: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:GPRS: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:GPRS:RFRX:RBER:ALL:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:GPRS`
	`: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. **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:GPRS: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:GPRS:RFRX:RBER:CIB:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS` `: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. **Note:** For further details regarding bit classes, refer to section voice coding.

Example	`:MEAS:GPRS:RFRX:RBER:CIB`
	`:CALC:GPRS: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:GPRS:RFRX:RBER:CIB:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS: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:GPRS:RFRX:RBER:CIB:LIM:STAT OFF Switches the limit check of the residual bit error ratio measurement for the class 1b bits off.

:CALCulate:GPRS:RFRX:RBER:CIB:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GPRS :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 lb** bits. **Note:** For further details regarding bit classes, refer to section voice coding.
Query	The query form of this command is not available.
Example	:CALC:GPRS:RFRX:RBER:CIB:LIM:UPP 5 Sets the upper limit for the class I bit error rate results to 5.0.

:CALCulate:GPRS:RFRX:RBER:CIB:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS :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 lb** bits. **Note:** For further details regarding bit classes, refer to section voice coding.
Query	The query form of this command is not available.
Example	:CALC:GPRS:RFRX:RBER:CIB:LIM:LOW 0.0 Sets the lower limit for the class I bit error rate results to 0.

:CALCulate:GPRS:RFRX:RBER:CII:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS :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. **Note:** For further details regarding bit classes, please refer to section voice coding.

Example	`:MEAS:GPRS:RFRX:RBER:CII`
	`:CALC:GPRS: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:GPRS:RFRX:RBER:CII:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS: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:GPRS:RFRX:RBER:CII:LIM:STAT ON` Switches the limit check of the residual bit error ratio measurement for the class II bits on.

:CALCulate:GPRS:RFRX:RBER:CII:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS` `: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. **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:GPRS:RFRX:RBER:CII:LIM:UPP 5` Sets the upper limit of the class II RBER measurement to 5%.

:CALCulate:GPRS:RFRX:RBER:CII:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:GPRS` `: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. **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:GPRS:RFRX:RBER:CII:LIM:LOW 0.0` Sets the lower limit of the class II RBER measurement to 0%.

:CALCulate:GPRS
:RFRX:RBER:FER:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS` `:RFRX:RBER:FER: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 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:GPRS:RFRX:RBER:FER`
	`:CALC:GPRS: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:GPRS:RFRX:RBER:FER:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS: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:GPRS:RFRX:RBER:FER:LIM:STAT ON` Switches the limit check of the frame erasure ratio measurement on.

:CALCulate:GPRS:RFRX:RBER:FER:LIMit:UPPer[:DATA]

Syntax	`:CALC:GPRS` `: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:GPRS:RFRX:RBER:FER:LIM:UPP 4.5` Sets the upper limit for the FER measurement results to 4.5%.

:CALCulate:GPRS:RFRX:RBER:FER:LIMit:LOWer[:DATA]

Syntax	`:CALC:GPRS` `: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. **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:GPRS:RFRX:RBER:FER:LIM:LOW 0` Sets the lower limit for the FER measurement results to 0%.

:CALCulate:GPRS:RFRX:BER:ALL:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS: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:GPRS:RFRX:BER:ALL` command.

Example	`:MEAS:GPRS:RFRX:BER:ALL 3`
	`:CALC:GPRS: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:GPRS:RFRX:BER:ALL:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS: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:GPRS:RFRX:BER:ALL:LIM:STAT OFF` Switches the limit check of all (nonresidual) bit error ratio measurements off.

:CALCulate:GPRS:RFRX:BER:ALL:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS`
	`: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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	`:CALC:GPRS: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:GPRS:RFRX:BER:ALL:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:GPRS`
	`:RFRX:BER:ALL:LIMit:LOWer[:DATA]`
	`<real1>,<real2>,<real3>`

Parameters	`realx` are three floating point real numbers given as a percentage. 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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	`:CALC:GPRS:RFRX:BER:ALL:LIM:LOW 0.0,0.0,0.0` Sets the lower limit for all three BER measurements (class Ia, class Ib and class II) to 0%.

:CALCulate:GPRS:RFRX:BER:CIA:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS: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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Example	:MEAS:GPRS:ARRay:RFRX:BER:CIA 5 :CALC:GPRS: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:GPRS:RFRX:BER:CIA:LIMit:STATe

Syntax	:CALCulate:GPRS: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:GPRS: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:GPRS:RFRX:BER:CIA:LIM:STAT ON Switches the limit check of the (nonresidual) bit error ratio measurement for the class Ia bits on.

:CALCulate:GPRS:RFRX:BER:CIA:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GPRS :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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	:CALC:GPRS:RFRX:BER:CIA:LIM:UPP 2 Sets the upper limit for bit error rate measurements on class Ia bits to 2.0%.

:CALCulate:GPRS:RFRX:BER:CIA:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS :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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	:CALC:GPRS:RFRX:BER:CIA:LIM:LOW 0.0 Sets the lower limit for BER measurements on class Ia bits to 0.

:CALCulate:GPRS:RFRX:BER:CIB:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS: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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Example	:MEAS:GPRS:RFRX:BER:CIB :CALC:GPRS: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:GPRS:RFRX:BER:CIB:LIMit:STATe

Syntax	:CALCulate:GPRS: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:GPRS: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:GPRS:RFRX:BER:CIB:LIM:STAT ON Switches the limit check of the (nonresidual) bit error ratio measurement for the class Ib bits on.

:CALCulate:GPRS:RFRX:BER:CIB:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GPRS :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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	:CALC:GPRS:RFRX:BER:CIB:LIM:UPP 2.0 Sets the upper limit for class Ib bits to 2%.

:CALCulate:GPRS:RFRX:BER:CIB:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS :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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	:CALC:GPRS:RFRX:BER:CIB:LIM:LOW 0 Sets the lower limit for class Ib bits to 0%.

:CALCulate:GPRS:RFRX:BER:CII:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS: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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Example	:MEAS:GPRS:ARRay:RFRX:BER:CII 3 :CALC:GPRS: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:GPRS:RFRX:BER:CII:LIMit:STATe

Syntax	:CALCulate:GPRS: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:GPRS: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:GPRS:RFRX:BER:CII:LIM:STAT ON Switches the limit check of the (nonresidual) bit error ratio measurement for the class II bits on.

:CALCulate:GPRS:RFRX:BER:CII:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GPRS :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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	:CALC:GPRS:RFRX:BER:CII:LIM:UPP 7.5 Sets the upper limit of the class II BER measurements to 7.5%.

:CALCulate:GPRS:RFRX:BER:CII:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS :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. **Note:** For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	:CALC:GPRS:RFRX:BER:CII:LIM:LOW 0 Sets the lower limit of the class II BER measurements to 0%.

:CALCulate:GPRS:RFRX:BER:CRC:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS:RFRX:BER:CRC:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.

Query	This command delivers back one boolean number, indicating, whether the result value of the (non-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. Note: For further details regarding bit classes, please refer to section voice coding.

Example	`:MEAS:GPRS:ARRay:RFRX:BER:CRC 3`
	`:CALC:GPRS:RFRX:BER:CII:LIM?`
	String returned in this example:`"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:GPRS:RFRX:BER:CRC:LIMit:STATe

Syntax	`:CALCulate:GPRS:RFRX:BER:CRC:LIMit:STATe`
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:GPRS: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:GPRS:RFRX:BER:CRC:LIM:STAT ON` Switches the limit check of the (non-residual) bit error ratio measurement for the class II bits on.

:CALCulate:GPRS
:RFRX:BER:CRC:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS` `:RFRX:BER:CRC: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 (non-residual) bit error ratio measurement of the class II bits. Note: For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	`:CALC:GPRS:RFRX:BER:CRC:LIM:UPP 7.5` Sets the upper limit of the class II BER measurements to 7.5%.

:CALCulate:GPRS
:RFRX:BER:CRC:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:GPRS` `:RFRX:BER:CRC: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 (non-residual) bit error ratio measurement of the class II bits. Note: For further details regarding bit classes, please refer to section voice coding.
Query	The query form of this command is not supported.
Example	`:CALC:GPRS:RFRX:BER:CRC:LIM:LOW 0` Sets the lower limit of the class II BER measurements to 0%.

:CALCulate:GPRS:RFRX:FBER:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS: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:GPRS:RFRX:FBER`
	 `:CALC:GPRS: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:GPRS:RFRX:FBER:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS:RFRX:FBER` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	`:CALC:GPRS:RFRX:FBER:LIM:STAT ON` Switches the limit check of the fast bit error ratio measurement on.

:CALCulate:GPRS:RFRX:FBER:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS:RFRX:FBER: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 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:GPRS:RFRX:FBER:LIM:UPP 7.5` Sets the upper limit of the FER measurement results to 7.5%.

:CALCulate:GPRS:RFRX:FBER:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:GPRS:RFRX:FBER: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 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:GPRS:RFRX:FBER:LIM:LOW 0.0` Sets the lower limit of the FER measurement results to 0%.

:CALCulate:GPRS:RFRX:BLER:BCS:LIMit[:FAIL]?

Syntax	`:CALCulate:GPRS:RFRX:BLER:BCS: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 block error rate measurement based on the block check sequence (BLER-BCS) 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:GPRS:ARRay:MULT:RFRX:BLER 3`
	`:CALC:GPRS:RFRX:BLER:BCS:LIM?`
	String returned:`"1"`
	This means that the block error rate based on the block check sequence was outside the limits.

:CALCulate:GPRS:RFRX:BLER:BCS:LIMit:STATe

Syntax	`:CALCulate:GPRS:RFRX:BLER:BCS: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:GPRS(:MULT):RFRX:BLER` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	`:CALC:GPRS:RFRX:BLER:BCS:LIM:STAT ON` Switches the limit check for the BCS-based block error rate measurement on.

:CALCulate:GPRS:RFRX:BLER:BCS:LIMit:UPPer[:DATA]

Syntax	:CALCulate:GPRS :RFRX:BLER:BCS:LIMit:UPPer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value is 0.0, the maximum is 100.0. The resolution is 0.1, the default value is 2.5.
Description	Sets the **upper** limit for the BCS-based block error rate measurement.
Query	The query form of this command is not supported.
Example	:CALC:GPRS:RFRX:BLER:BCS:LIM:UPP 2 Sets the upper limit for BCS-based block error rate measurements to 2.0%.

:CALCulate:GPRS:RFRX:BLER:BCS:LIMit:LOWer[:DATA]

Syntax	:CALCulate:GPRS :RFRX:BLER:BCS: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 BCS-based block error rate measurement.
Query	The query form of this command is not supported.
Example	:CALC:GPRS:RFRX:BLER:BCS:LIM:LOW 0.0 Sets the lower limit for BCS-based BLER measurements to 0%.

:CALCulate:GPRS:RFRX:BLER:USF:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS:RFRX:BLER:USF: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 block error rate measurement based on the uplink state flag (BLER-USF) 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:GPRS:ARRay:RFRX:BLER 3 :CALC:GPRS:RFRX:BLER:USF:LIM? String returned:"1" This means that the block error rate based on the block check sequence was outside the limits.

:CALCulate:GPRS:RFRX:BLER:USF:LIMit:STATe

Syntax	:CALCulate:GPRS:RFRX:BLER:USF: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:GPRS(:MULT):RFRX:BLER` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	`:CALC:GPRS:RFRX:BLER:USF:LIM:STAT ON` Switches the limit check for the USF-based block error rate measurement on.

:CALCulate:GPRS:RFRX:BLER:USF:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS` `:RFRX:BLER:USF:LIMit:UPPer[:DATA] <real1>`
Parameters	`real1` is a floating point real number. The minimum value is `0.0`, the maximum is `100.0`. The resolution is `0.1`, the default value is `2.5`.
Description	Sets the **upper** limit for the USF-based block error rate measurement.
Query	The query form of this command is not supported.
Example	`:CALC:GPRS:RFRX:BLER:USF:LIM:UPP 2` Sets the upper limit for USF-based block error rate measurements to 2.0%.

:CALCulate:GPRS:RFRX:BLER:USF:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:GPRS` `:RFRX:BLER:USF: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 USF-based block error rate measurement.
Query	The query form of this command is not supported.
Example	`:CALC:GPRS:RFRX:BLER:USF:LIM:LOW 0.0` Sets the lower limit for USF-based BLER measurements to 0%.

:CALCulate:GPRS:RFSPectrum:ACPM:MAVerage?

Syntax	`:CALCulate:GPRS: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:GPRS:ARR:RFSP:ACPM:MOD`
	`:CALC:GPRS:RFSP:ACPM:MAV?`

:CALCulate:GPRS:RFSPectrum:ACPM:MMINimum?

Syntax	:CALCulate:GPRS: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:GPRS:ARR:RFSP:ACPM:MOD :CALC:GPRS:RFSP:ACPM:MMIN?

:CALCulate:GPRS:RFSPectrum:ACPM:MMAXimum?

Syntax	:CALCulate:GPRS: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:GPRS:ARR:RFSP:ACPM:MOD :CALC:GPRS:RFSP:ACPM:MMAX?

:CALCulate:GPRS:RFSPectrum:ACPM:MSIGma?

Syntax	:CALCulate:GPRS: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, 23 pairs of mean values and standard deviation values are returned. The values are floating point real values and separated by commas.
Example	:CALC:RES :MEAS:GPRS:ARR:RFSP:ACPM:MOD ...(other SCPI commands)... :CALC:GPRS:RFSP:ACPM:MSIG?

:CALCulate:GPRS
:RFSPectrum:ACPM:MODulation:LIMit[:FAIL]?

Syntax	:CALCulate:GPRS: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:GPRS: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:GPRS:RFSP:ACPM:MOD measurement will be performed on ($\pm$)100 kHz and ($\pm$)250 kHz offset from the carrier as well as on all multiples of ($\pm$)200 kHz offset from the carrier up to (and including) an offset of ($\pm$)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, 0,0,0,0,0,0,0,0,0,0,0,0,0,0" 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	:CALCulate:GSM:RFSPectrum :ACPM:MODulation: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:GPRS: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:GPRS:RFSP:ACPM:MOD:LIM:STAT ON Switches the limit check of the ACPM of the modulation spectrum measurement on.

**:CALCulate:GSM:RFSP:RFSPectrum
:ACPM:MODulation:LIMit:UPPer[:DATA]**

Syntax	`:CALCulate:GSM:RFSP: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:GPRS: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	`:CALCulate: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:GPRS: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
:GPRS:RFSPectrum:ACPM:TRANsient:LIMit[:FAIL]?

Syntax	`:CALCulate` `:GPRS: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:GPRS: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:GPRS:ARRay:RFSP:ACPM:TRANsient 10`
	`:CALC:GPRS: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"`
	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:GPRS:RFSPectrum
:ACPM:TRANsient:LIMit:STATe

Syntax	`:CALCulate:GPRS: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:GPRS: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:GPRS:RFSP:ACPM:TRAN:LIM:STAT ON` Switches the limit check of the ACPM of the spectrum due to switching transients on.

:CALCulate:GPRS:RFSPectrum :ACPM:TRANsient:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:GPRS: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 is not available.
Example	`:CALC:GPRS:RFSP:ACPM:TRAN:LIM:UPP?` This command will return the 27 upper limit values.

:CALC:GPRS:RFSP:ACPM:TRAN:LIMit:LOWer[:DATA]

Syntax	`:CALC:GPRS`
	`: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 is not available.
Example	`:CALC:GPRS:RFSP:ACPM:TRAN:LIM:LOW?` This command will return the 27 lower limit values.

:CALCulate:GPRS:RFSPectrum:MSPectrum:VALue

Syntax	`:CALCulate:GPRS:RFSPectrum:MSPectrum:VALue <real1>`
Parameters	`real1` is a floating point real number. The minimum value is `-2,000,000.0`, the maximum is `2,000,000.0`. 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 back 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:GPRS:RFSP:MSP:VAL` command is not really a measurement - it is an interpolation of the spectrum curve for the frequency specified.
Example	`:MEAS:GPRS: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:GPRS:RFSPectrum:MSPectrum:MAXPeak

Syntax	`:CALCulate:GPRS: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 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:GPRS:RFSP:MSP:MAXP` command is not really a measurement - it is an readout of the spectrum curve.
Example	`:MEAS:GPRS:BLOC:MSP:AVG 20` `:CALC:GPRS: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"`.

:CALC:EGPRS:RFTX:MAVerage

Syntax	`:CALCulate:EGPRs: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:EGPR:MEAS:GRO:RFTX POW,ERMS`
	`:CALC:RES`
	`:MEAS:EGPR:ARR:RFTX:GRO 20`
	`:CALC:EGPR:RFTX:MAV?`
	In this example, first the group of measurements is defined by a power measurement combined with a EVM 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:
	`"4.53,5.98"`.

:CALCulate:EGPRs:RFTX:MMINimum

Syntax	`:CALCulate:EGPRs: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:EGPR:MEAS:GRO:RFTX POW,ERMS`
	`:CALC:RES`
	`:MEAS:EGPR:ARR:RFTX:GRO 20`
	`:CALC:EGPR:RFTX:MMIN?`
	In this example, first the group of measurements is defined by a power measurement combined with a EVM 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:
	`"3.53,2.98"`.

:CALCulate:EGPRs:RFTX:MMAXimum

Syntax	`:CALCulate:EGPRs: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	`:CONF:EGPR:MEAS:GRO:RFTX POW,ERMS`
	`:CALC:RES`
	`:MEAS:EGPR:ARR:RFTX:GRO 20`
	`:CALC:EGPR:RFTX:MMAX?`
	In this example, first the group of measurements is defined by a power measurement combined with a EVM 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:
	`"6.53,15.98"`.

`:CALCulate:EGPRs:RFTX:MSIGma`

Syntax	`:CALCulate:EGPRs: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 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 are separated by commas.

Example	`:MEAS:EGPRs:RFTX:FREQ`
	`:CALC:RES`
	`...(other SCPI commands)...`
	`:CALC:EGPR:RFTX:MSIG?`
	In this example, first a continuous measurement of the mobile's frequency error is started. After some time, the :CALC:EGPR: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:EGPRs:RFTX:ALL:LIMit[:FAIL]?`

Syntax	`:CALCulate:EGPRs:RFTX:ALL:LIMit[:FAIL]?`
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command delivers 8 boolean numbers in a string, separated by commas. These numbers indicate, whether the single measurement results of the :MEAS:EGPRs[: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. error vector magnitude rms-averaged measurement, 2. maximum error vector magnitude measurement, 3. error vector magnitude that is not exceede by 95% of the symbols of a burst measurement, 4. error vector magnitude origin offset measurement, 5. measurement of the mobile's frequency error, 6. measurement of the burst lenght, 7. timing error measurement, 8. measurement of the mobile's RF output power level.

Example	`:MEAS:EGPR:RFTX:ALL`
	`:CALC:EGPR:RFTX:ALL:LIM?`
	String returned in this example:`"0,0,0,0,0,0,0,1"`
	1 on position 8 indicates that the mobile's RF power level violated its test limits.

`:CALCulate:EGPRs:RFTX:ALL:LIMit:STATe`

Syntax	`:CALCulate:EGPRs: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 of the relevant `EGPRS RF TX` measurement result values of the `MEAS:EGPRs:RFTX:ALL` measurement against their user-definable limits either `ON` or `OFF`.
Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:ALL:LIM:STAT OFF`
	Switches the limit check for the most relevant EGPRS RF TX measurements off.

`:CALCulate:EGPRs:RFTX:ALL:LIMit:UPPer[:DATA]`

Syntax	`:CALCulate:EGPRs:RFTX:ALL:LIMit:UPPer[:DATA]` `<real1>,<real2>,<real3>,<real4>,<real5>,<real6>,` `<real7>,<real8>`
Parameters	`realx` are eight floating point real numbers.
	`real1`: The minimum value is `0.0`, the maximum is `500.0`. The minimum resolution possible is `0.1`, the default value is `9.0`.
	`real2`: The minimum value is `0.0`, the maximum is `500.0`. The minimum resolution possible is `0.1`, the default value is `30.0`.
	`real3`: The minimum value is `0.0`, the maximum is `500.0`. The minimum resolution possible is `1.0`, the default value is `15.0`.
	`real4`: The minimum value is `-120.0`, the maximum is `0.0`. The minimum resolution possible is `0.1`, the default value is `-30.0`.
	`real5`: The minimum value is `0.0`, the maximum is `100,000.0`. The minimum resolution possible is `1`, the default value is `90.0`.
	`real6`: The minimum value is `0.0`, the maximum is `700.0`. The minimum resolution possible is `0.1`, the default value is `562.8`.
	`real7`: The minimum value is `0.0`, the maximum is `64.0`. The minimum resolution possible is `0.01`, the default value is `3.0`.
	`real8`: 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 eight measurement types with one command. The order of the floating point real figures defines their meaning:
	`real1` represents the upper limit of the error vector magnitude rms-averaged measurement; the physical dimension of the number stated is %.
	`real2` represents the upper limit of the maximum error vector magnitude measurement ; the physical dimension of the number stated is %.
	`real3` represents the upper limit of the error vector magnitude that is not exceede by 95% of the symbols of a burst measurement; the physical dimension of the number stated is %.
	`real4` represents the upper limit of the error vector magnitude origin offset measurement; the physical dimension of the number stated is dBc.
	`real5` represents the upper limit of the frequency error measurement; the physical dimension of the number stated is Hertz.
	`real6` represents the upper limit of the measurement of the burst length; the physical dimension of the number stated is microseconds.
	`real7` 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.
	`real8` represents the upper limit of the RF power level measurement; the physical dimension of the number stated is dBm.
Query	The query form of this command is not supported.
Example	`:CALC:EGPR:RFTX:ALL:LIM:UPP 12,40,20,-20,90,562.8,0,13`
	The command sets the upper limits for the EGPRS transmitter measurement to "`12.0,40.0,20.0,-20.0,90.0,562.8,0.0,13.0`", which could be the limits for the next `EGPRS RFTX:ALL` measurements.

:CALCulate:EGPRs:RFTX:ALL:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:EGPRs:RFTX:ALL:LIMit:LOWer[:DATA]`
	`<real1>,<real2>,<real3>,<real4>,<real5>,<real6>,`
	`<real7>,<real8>`

Parameters	`realx` are eight floating point real numbers.
	`real1`: The minimum value is `0.0`, the maximum is `500.0`. The minimum resolution possible is `0.1`, the default value is `0.0`.
	`real2`: The minimum value is `0.0`, the maximum is `500.0`. The minimum resolution possible is `0.1`, the default value is `0.0`.
	`real3`: The minimum value is `0.0`, the maximum is `500.0`. The minimum resolution possible is `1.0`, the default value is `0.0`.
	`real4`: The minimum value is `-120.0`, the maximum is `0.0`. The minimum resolution possible is `0.1`, the default value is `-120.0`.
	`real5`: The minimum value is `-100,000.0`, the maximum is `0.0`. The minimum resolution possible is `1`, the default value is `-90.0`.
	`real6`: The minimum value is `0.0`, the maximum is `700.0`. The minimum resolution possible is `0.1`, the default value is `542.8`.
	`real7`: The minimum value is `-64.0`, the maximum is `0.0`. The minimum resolution possible is `0.01`, the default value is `-3.0`.
	`real8`: The minimum value is `-99.0`, the maximum is `0.0`. The minimum resolution possible is `0.01`, the default value is `-2.0`.

Description	Sets the **lower** limits for eight measurement types with one command. The order of the floating point real figures defines their meaning: `real1` represents the lower limit of the error vector magnitude rms-averaged measurement; the physical dimension of the number stated is %. `real2` represents the lower limit of the maximum error vector magnitude measurement ; the physical dimension of the number stated is %. `real3` represents the lower limit of the error vector magnitude that is not exceede by 95% of the symbols of a burst measurement; the physical dimension of the number stated is %. `real4` represents the lower limit of the error vector magnitude origin offset measurement; the physical dimension of the number stated is dBc. `real5` represents the lower limit of the frequency error measurement; the physical dimension of the number stated is Hertz. `real6` represents the lower limit of the measurement of the burst length; the physical dimension of the number stated is microseconds. `real7` 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. `real8` 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:EGPR:RFTX:ALL:LIM:LOW 0,0,0,-80,-90,542.8,0,0` The command sets the lower limits for the EGPRS transmitter measurement to `"0.0,0.0,0.0,-80.0,-90.0,542.8,0.0,0.0"`, which could be the limits for the next EGPRS `RFTX:ALL` measurements.

`:CALCulate:EGPRs:RFTX:ERMS:LIMit[:FAIL]`

Syntax	`:CALCulate:EGPRs:RFTX:ERMS:LIMit[:FAIL]?`
Parameters	There are no parameters with this command.
Description	There is solely a query form of this command available.
Query	Returns a boolean value indicating whether the RMS EVM measurement result has failed the limits.
Example	`:CALC:EGPR:ERMS:LIM:FAIL?` Returns **`0`** after an RMS EVM measurement failing the set limits.

`:CALCulate:EGPRs:RFTX:ERMS:LIMit:STATe`

Syntax	`:CALCulate:EGPRs:RFTX:ERMS: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:EGPRs:RFTX:ERMS` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:ERMS:LIM:STAT ON` Switches the limit check on RMS EVM measurements on.

:CALCulate:EGPRs:RFTX:ERMS:LIMit:UPPer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:ERMS:LIMit:UPPer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 500. The resolution is 0.1 and the default value is 9.
Description	Sets the upper limit for the RMS EVM measurement (in %).
Query	There is only a command form available.
Example	:CALC:EGPR:RFTX:ERMS:LIM:UPP 45 This command will sets the upper limit to 45%.

:CALCulate:EGPRs:RFTX:ERMS:LIMit:LOWer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:ERMS:LIMit:LOWer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 500. The resolution is 0.1 and the default value is 0.
Description	Sets the lower limit for the RMS EVM measurement (in %).
Query	There is only a command form available.
Example	:CALC:EGPR:RFTX:ERMS:LIM:LOW 0.8 Sets the lower limit of the error vector magnitude rms-averaged measurement result to 0.8 %.

:CALCulate:EGPRs:RFTX:EPEAk:LIMit[:FAIL]

Syntax	:CALCulate:EGPRs:RFTX:EPEAk:LIMit[:FAIL]?
Parameters	There are no parameters with this command.
Description	There is solely a query form of this command available.
Query	Returns a boolean value indicating whether the peak EVM measurement result has failed the limits.
Example	:MEAS:EGPR:ARR:RFTX:EPEA 20 :CALC:EGPR:RFTX:EPEA:LIM? String returned in this example:"0" This string delivered back indicates that there was no violation of the measurement limits.

:CALCulate:EGPRs:RFTX:EPEAk:LIMit:STATe

Syntax	:CALCulate:EGPRs:RFTX: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 MEAS:EGPRs:RFTX:EVM:PEAk measurement against their user-definable limits either on or off.

Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:EPEA:LIM:STAT ON` Switches the limit check on peak EVM measurements on.

`:CALCulate:EGPRS:RFTX:EPEAk:LIMit:UPPer[:DATA]`

Syntax	`:CALCulate:EGPRS:RFTX:EPEAk:LIMit:UPPer[:DATA] <real1>`
Parameters	`real1` is a floating point real number. The minimum value for each is `0`, the maximum is `500`. The resolution is `0.1` and the default value is `30`.
Description	Sets the upper limit for the peak EVM measurement (in %).
Query	There is only a command form available.
Example	`:CALC:EGPR:RFTX:EPEA:LIM:UPP 25.0` This command will sets the upper limit to 25%.

`:CALCulate:EGPRS:RFTX:EPEAk:LIMit:LOWer[:DATA]`

Syntax	`:CALCulate:EGPRS:RFTX:EPEAk:LIMit:LOWer[:DATA] <real1>`
Parameters	`real1` is a floating point real number. The minimum value for each is `0`, the maximum is `500`. The resolution is `0.1` and the default value is `0`.
Description	Sets the lower limit for the peak EVM measurement (in %).
Query	There is only a command form available.
Example	`:CALC:EGPR:RFTX:EPEAk:LIM:LOW 0` This command will sets the lower limit to 0%.

`:CALCulate:EGPRs:RFTX:ENFTh:LIMit[:FAIL]?`

Syntax	`:CALCulate:EGPRs:RFTX:ENFTh:LIMit[:FAIL]?`
Parameters	There are no parameters with this command.
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 error vector magnitude that is not exceed by 95% of the symbols of a burst 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:EGPR:ARR:RFTX:ENFT 20`
	`:CALC:EGPR:RFTX:ENFT:LIM?`
	String returned in this example: "0"
	This string delivered back indicates that there was no violation of the measurement limits.

:CALCulate:EGPRs:RFTX:ENFTh:LIMit:STATe

Syntax	:CALCulate:EGPRs:RFTX:ENFTh: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:EGPRs:RFTX:ENFTh measurement against their user-definable limits either ON or OFF.
Query	There is no query form of this command available.
Example	:CALC:EGPR:RFTX:ENFT:LIM:STAT ON Switches the limit check for the error vector magnitude that is not exceede by 95% of the symbols of a burst measurement on.

:CALCulate:EGPRs:RFTX:ENFTh:LIMit:UPPer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:ENFTh:LIMit:UPPer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 500. The resolution is 0.1 and the default value is 15.
Description	Sets the **upper** limit of the error vector magnitude that is not exceeded by 95% of the symbols of a burst measurement; the physical dimension of the number stated is %.
Query	There is no query form of this command available.
Example	:CALC:EGPR:RFTX:ENFT:LIMi:UPP 75 Sets the upper limit of the error vector magnitude that is not exceeded by 95% of the symbols of a burst measurement result to 75 %.

:CALCulate:EGPRs:RFTX:ENFTh:LIMit:LOWer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:ENFTh:LIMit:LOWer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 500. The resolution is 0.1 and the default value is 0.
Description	Sets the lower limit of the error vector magnitude that is not exceeded by 95% of the symbols of a burst measurement; the physical dimension of the number stated is %.
Query	There is no query form of this command available.
Example	:CALC:EGPR:RFTX:ENFT:LIM:LOW 1.1 Sets the lower limit of the error vector magnitude that is not exceede by 95% of the symbols of a burst measurement result to 1.1 %.

:CALCulate:EGPRs:RFTX:EOFFset:LIMit[:FAIL]?

Syntax	:CALCulate:EGPRs:RFTX:EOFFset:LIMit[:FAIL]?
Parameters	There are no parameters with this command.
Description	There is solely a query form of this command available.

4463 GSM/GPRS System Option and 4468 EDGE System Option   Version 6.20

Query	This command delivers one boolean number, indicating whether the user-definable limits of the error vector magnitude origin offset 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:EGPR:ARR:RFTX:EOFF 20`
	`:CALC:EGPR:RFTX:EOFF:LIM?`
	String returned in this example: "0"
	This string delivered back indicates that there was no violation of the measurement limits.

`:CALC:EGPRs:RFTX:EOFFset:LIMit:STATe`

Syntax	`:CALC:EGPRs:RFTX:EOFFset: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:EGPRs:RFTX:EOFFSet` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:EOFF:LIM:STAT ON`
	Switches the limit check on origin offset measurements on.

`:CALCulate:EGPRs:RFTX:EOFFset:LIMit:UPPer[:DATA]`

Syntax	`:CALCulate:EGPRs:RFTX:EOFFset:LIMit:UPPer[:DATA] <real1>`
Parameters	`real1` is a floating point real number. The minimum value for each is -120, the maximum is 0. The resolution is 0.1 and the default value is -30.
Description	Sets the **upper** limit of the error vector magnitude origin offset measurement; the physical dimension of the number stated is dBc.
Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:EOFF:LIM:UPP -10`
	Sets the upper limit of the error vector magnitude origin offset measurement result to -10 dBc.

`:CALCulate:EGPRs:RFTX:EOFFset:LIMit:LOWer[:DATA]`

Syntax	`:CALCulate:EGPRs:RFTX:EOFFset:LIMit:LOWer[:DATA] <real1>`
Parameters	`real1` is a floating point real number. The minimum value for each is -120, the maximum is 0. The resolution is 0.1 and the default value is -120.
Description	Sets the lower limit of the error vector magnitude origin offset measurement; the physical dimension of the number stated is dBc.

Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:EOFF:LIM:LOW -100` Sets the lower limit of the error vector magnitude origin offset measurement result to `-100 dBc`.

:CALCulate:EGPRs:RFTX:FREQuency:LIMit[:FAIL]?

Syntax	`:CALCulate:EGPRs:RFTX:FREQuency:LIMit[:FAIL]?`
Parameters	There are no parameters with this command.
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:EGPR:ARR:RFTX:FREQ 20`
	`:CALC:EGPR:RFTX:FREQ:LIM?`
	String returned in this example:"0"
	This string delivered back indicates that there was no violation of the measurement limits.

:CALCulate:EGPRs:RFTX:FREQuency:LIMit:STATe

Syntax	`:CALCulate:EGPRs: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:EGPRs:RFTX:FREQuency` measurement against their user-definable limits either `ON` or `OFF`.
Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:FREQuency:LIM:STAT ON` Switches the limit check on frequency measurements on.

:CALCulate:EGPRs:RFTX:FREQuency:LIMit:UPPer[:DATA]

Syntax	`:CALCulate:EGPRs:RFTX:FREQuency:LIMit:UPPer[:DATA]` `<real1>`
Parameters	`real1` is a floating point real number. The minimum value for each is `0`, the maximum is `100000`. The resolution is `1` and the default value is `90`.
Description	Sets the upper limit for the frequency error measurement (in Hertz).
Query	There is only a command form available.

Example	:CALC:EGPR:RFTX:FREQ:LIM:UPP 45
	Sets the upper limit of the frequency error measurement result to 45 Hz.

:CALCulate:EGPRs:RFTX:FREQuency:LIMit:LOWer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:FREQuency:LIMit:LOWer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is –100000, the maximum is 0. The resolution is 1 and the default value is –90.
Description	Sets the lower limit for the frequency error measurement (in Hertz).
Query	There is only a command form available.
Example	:CALC:EGPR:RFTX:FREQ:LIM:LOW –60.8
	Sets the lower limit of the frequency error measurement result to –60.8 Hz.

:CALCulate:EGPRs:RFTX:FREQuency:LIMit:GSM

Syntax	:CALCulate:EGPRs:RFTX:FREQuency:LIMit:GSM <real1>
Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 100000. The resolution is 1 and the default value is 90.
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:EGPR: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:EGPRs:RFTX:FREQuency:LIMit:PCN

Syntax	:CALCulate:EGPRs:RFTX:FREQuency:LIMit:PCN <real1>
Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 100000. The resolution is 1 and the default value is 180.
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 only a command form available.
Example	:CALC:EGPR: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:EGPRs:RFTX:FREQuency:LIMit:PCS

Syntax	:CALCulate:EGPRs:RFTX:FREQuency:LIMit:PCS <real1>

Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 100000. The resolution is 1 and the default value is 180.
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 only a command form available.
Example	:CALC:EGPR: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:EGPRS:RFTX:LENGth:LIMit[:FAIL]

Syntax	:CALCulate:EGPRS: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:EGPR:ARR:RFTX:LENG 10 :CALC:EGPR:RFTX:LENG:LIM? String returned in this example: "0" This string indicates that there was no violation of the measurement limits.

:CALCulate:EGPRs:RFTX:LENGth:LIMit:STATe

Syntax	:CALCulate:EGPRs: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:EGPRs:RFTX:LENGth measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	:CALC:EGPR:RFTX:LENG:LIM:STAT ON Switches the limit check on frequency measurements on.

:CALCulate:EGPRs:RFTX:LENGth:LIMit:UPPer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:LENGth:LIMit:UPPer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 700. The resolution is 0.1 and the default value is 562.8.

Description	Sets the upper limit for the burst length measurement; the physical dimension of the number stated is microseconds.
Query	There is only a command form available.
Example	`:CALC:EGPR:RFTX:LENG:LIM:UPP 563` This command will sets the upper limit of the burst length to 563 µs.

:CALCulate:EGPRs:RFTX:LENGth:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:EGPRs:RFTX:LENGth:LIMit:LOWer[:DATA] <real1>`
Parameters	`real1` is a floating point real number. The minimum value for each is 0, the maximum is `700`. The resolution is `0.1` and the default value is `542.8`.
Description	Sets the lower limit for the burst length measurement; the physical dimension of the number stated is microseconds.
Query	There is only a command form available.
Example	`:CALC:EGPR:RFTX:LENG:LIM:LOW 541.9` This command will sets the lower limit of the burst length to 541.9 µs.

:CALCulate:EGPRS:RFTX:UTIMe:LIMit[:FAIL]

Syntax	`:CALCulate:EGPRS: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:EGPR:ARR:RFTX:UTIM 10`
	 `:CALC:EGPR:RFTX:UTIM:LIM?`
	 String returned in this example: `"0"`
	 This string indicates that there was no violation of the measurement limits.

:CALCulate:EGPRs:RFTX:UTIMe:LIMit:STATe

Syntax	`:CALCulate:EGPRs: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:EGPRs:RFTX:UTIMe` measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:UTIM:LIM:STAT ON` Switches the limit check for the uplink timing error measurement on.

:CALCulate:EGPRs:RFTX:UTIMe:LIMit:UPPer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:UTIMe:LIMit:UPPer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 64. The resolution is 0.01 and the default value is 3.
Description	Sets the upper limit for the uplink 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 only a command form available.
Example	:CALC:EGPR:RFTX:UTIM:LIM:UPP 5 Sets the upper limit of the uplink timing error measurement result to 5 μs.

:CALCulate:EGPRs:RFTX:UTIMe:LIMit:LOWer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:UTIMe:LIMit:LOWer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is –64, the maximum is 0. The resolution is 0.01 and the default value is –3.
Description	Sets the lower limit for the uplink timing error measurement; the physical dimension of the number stated is microseconds.
Query	There is only a command form available.
Example	:CALC:EGPR:RFTX:UTIM:LIM:LOW –5 Sets the lower limit of the uplink error timing measurement to –5 μs.

:CALCulate:EGPRS:RFTX:POWer:LIMit[:FAIL]

Syntax	:CALCulate:EGPRS: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:EGPR:ARR:POW 20 :CALC:EGPR:RFTX:POW:LIM? String returned in this example: "0" This string indicates that there was no violation of the measurement limits.

:CALCulate:EGPRs:RFTX:POWer:LIMit:STATe

Syntax	:CALCulate:EGPRs: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:EGPRs:RFTX:POWer measurement against their user-definable limits either on or off.
Query	There is no query form of this command available.
Example	:CALC:EGPR:RFTX:POWer:LIM:STAT ON Switches the limit check for the RF output power level measurement on.

:CALCulate:EGPRs:RFTX:POWer:LIMit:UPPer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:POWer:LIMit:UPPer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is 0, the maximum is 99. The resolution is 0.01 and 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 only a command form available.
Example	:CALC:EGPR:RFTX:POW:LIM:UPP 13 Sets the upper limit of the peak power measurement to 13 µs.

:CALCulate:EGPRs:RFTX:POWer:LIMit:LOWer[:DATA]

Syntax	:CALCulate:EGPRs:RFTX:POWer:LIMit:LOWer[:DATA] <real1>
Parameters	real1 is a floating point real number. The minimum value for each is −99, the maximum is 0. The resolution is 0.01 and 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 only a command form available.
Example	:CALC:EGPR:RFTX:POW:LIM:LOW −45 Sets the lower limit of the peak power measurement to −13 dBm.

:CALCulate:EGPRs:RFTX:POWer:LIMit:GSM

Syntax	:CALCulate:EGPRs:RFTX:POWer:LIMit:GSM <real1>,<real2>,...,<real18>
Parameters	real1 thru real18 are floating point real numbers. The minimum value for each is 0, the maximum is 30 and the resolution is 0.1. The default values are 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, respectively.

Description	Sets the **symmetrical** limits of the peak power for measurements in synchronous (call) mode for GSM 900 and GSM 850; the physical dimension of the number stated is dB. The first value applies to power control steps 0 to 2 (39 dBm), the second value applies to power control level 3 (37 dBm) and so forth up to the eighteenth value for power control steps 19 thru 31 (5 dBm).
Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:POW:LIM:GSM 2, 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, 5, 5, 5, 5 dB around the nominal level.

`:CALCulate:EGPRs:RFTX:POWer:LIMit:PCN`

Syntax	`:CALCulate:EGPRs:RFTX:POWer:LIMit:PCN` `<real1>,<real2>,...,<real19>`
Parameters	`real1` thru `real19` are floating point real numbers. The minimum value for each is `0`, the maximum is `30` and the resolution 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`, respectively.
Description	Sets the **symmetrical** limits of the peak power for measurements in synchronous (call) mode for GSM 1800 (also known as DCS-1800 or PCN); 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 only a command form available.
Example	`:CALC:EGPR: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 level 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:EGPRs:RFTX:POWer:LIMit:PCS`

Syntax	`:CALCulate:EGPRs:RFTX:POWer:LIMit:PCS` `<real1>,<real2>,...,<real18>`
Parameters	`real1` through `real18` are floating point real numbers. The minimum value for each is `0`, the maximum is `30` and the resolution is `0.1`. The default values are `2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5`, respectively.
Description	Sets the **symmetrical** limits of the peak power for measurements in synchronous (call) mode for GSM 1900 (also known as PCS); 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 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 thru 28 (0 dBm).
Query	There is only a command form available.

| Example | :CALC:EGPR:RFTX:POW:LIM:PCS 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5 |
| | Sets the lower limit of the power measurement result on GSM 1900 channels to 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5 dB around the nominal level. |

:CALCulate:EGPRs:RFTX:TEMPlate[:NORMal]:LIMit[:FAIL]?

Syntax	:CALCulate:EGPRs:RFTX:TEMPlate[:NORMal]:LIMit[:FAIL]?
Parameters	There are no parameters.
Description	There is solely a query form of this command available.
Query	This command returns 18 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 18 boolean numbers correspond to the limits of the PTT as defined on the user interface or with the :CALC:EGPRs:RFTX:TEMP commands as described below. The first eleven numbers stand for the eleven 'upper' segments of the burst, while the following seven numbers stand for the seven 'lower' segments of the burst. Refer to section Power-Time Templates (PTT) for further details.
Example	:MEAS:EGPR:RFTX:TEMP
	:CALC:EGPR:RFTX:TEMP:LIM?
	String returned in this example:
	"0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0"
	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:EGPRs:RFTX:TEMPlate[:NORMal]:LIMit:AUTO

Syntax	:CALCulate:EGPRs: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 4400 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 4400 will use the manually set limits instead (see description of the :CALC:EGPRs:RFTX:TEMP:LIM:UPP and :CALC:EGPRs:RFTX:TEMP:LIM:LOW commands for further details).
Query	There is no query form of this command available.
Example	:CALC:EGPR:RFTX:TEMP:LIM:AUTO ON
	Switches the limit check for the burst shape to use the standard PTTs as defined by ETSI standards.

:CALCulate:EGPRs:RFTX:TEMPlate[:NORMal]:LIMit:STAT

Syntax	:CALCulate:EGPRs:RFTX:TEMPlate[:NORMal]:LIMit:STAT <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:EGPRs:RFTX:TEMPlate measurement against their user-definable limits either ON or OFF.
Query	There is no query form of this command available.
Example	:CALC:EGPR:RFTX:TEMP:LIM:STAT ON Switches on the limit check of the burst shape against the PTT for normal bursts.

:CALCulate:EGPRs
:RFTX:TEMPlate[:NORMal]:LIMit:UPPer[:DATA]

Syntax	:CALCulate:EGPRs :RFTX:TEMPlate[:NORMal]:LIMit:UPPer[:DATA] <real0t>,<real0p>,<real1t>,<real1p>, <real2t>,<real2p>,<real3t>,<real3p>, <real4t>,<real4p>,<real5t>,<real5p>, <real6t>,<real6p>,<real7t>,<real7p>, <real8t>,<real8p>, <real9t>,<real9p>, <real10t>,<real10p>, <real11t>,<real11p>

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`: `4`, for `real6t`: `539.0`, for `real7t`: `543.0`, for `real8t`: `553.0`, for `real9t`: `561.0`, for `real10t`: `571.0`, for `real11t`: `580.0`. The minimum value for `realxp` is `-150.0`, the maximum value `5.0`. The minimum resolution possible for `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`: `2.4`, for `real6p`: `4.0`, for `real7p`: `2.4`, for `real8p`: `4.0`, for `real9p`: `-6.0`, for `real10p`: `-30.0`, for `real11p`: `-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 twelve 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:EGPR:RFTX:TEMP:LIM:UPP -41,-59,-28,-59,-18,-30,-10,-6,0,4,4,2.4,539,4,543,2.4,553,4,561,-6,571,-30,580,-59`

:CALCulate:EGPRs
:RFTX:TEMPlate[:NORMal]:LIMit:LOWer[:DATA]

Syntax	`:CALCulate:EGPRs`
	`:RFTX:TEMPlate[:NORMal]:LIMit:LOWer[:DATA]`
	`<real0t>,<real0p>,<real1t>,<real1p>,`
	`<real2t>,<real2p>,<real3t>,<real3p>,`
	`<real4t>,<real4p>,<real5t>,<real5p>, <real6t>,<real6p>,`
	`<real7t>,<real7p>`

Parameters	`realxt` and `realxp` are floating point real numbers. The minimum value for `realxt` is `-41.0`, the maximum value `580.0`. The minimum resolution possible for `realxt` is `0.1`. The default values are: for `real0t`: `-41.0`, for `real1t`: `0.0`, for `real2t`: `2.0`, for `real3t`: `4.0`, for `real4t`: `539.0`, for `real5t`: `541.0`, for `real6t`: `543.0`, for `real7t`: `580.0`. The minimum value for `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`: `-2.0`, for `real3p`: `0.0`, for `real4p`: `-20.0`, for `real5p`: `-0.0`, for `real6p`: `-2.0`, for `real7p`: `-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 eight 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:EGPR:RFTX:TEMP:LIM:LOW` `-41,-150,0,-150,0,-40,20, -1,270,-1,543,-1,543,-150`

`:CALCulate:EGPRs`
`:RFTX:CORNer[:NORMal]:POSition`

Syntax	`:CALCulate:EGPRs`
	`:RFTX:CORNer[:NORMal]:POSition`
	`<real1>,<real2>,<real3>,<real4>,`
	`<real5>,<real6>,<real7>,<real8>`

Parameters	`realx` stands for floating point real numbers. The minimum value for `realx` is `-28.0`, the maximum value `580.0`. The minimum resolution possible for `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:EGPR:RFTX:CORN:POS -28,-18,-10,0,542.8,552.8,562.8,572.8`

:CALCulate:EGPRs
:RFTX:CORNer[:NORMal]:LIMit[:FAIL]?

Syntax	`:CALCulate:EGPRs` `: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:EGPRs:RFTX:CORN:NORM:POS` command as described above. Please refer to section Corner Points for further details.
Example	`:MEAS:EGPR:ARR:CORN 10` `:CALC:EGPR:RFTX:CORN:LIM?` 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:EGPRs
:RFTX:CORNer[:NORMal]:LIMit:STATe

Syntax	`:CALCulate:EGPRs` `: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:EGPRs:RFTX:CORNer` measurement against their user-definable limits either ON or OFF.
Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFTX:CORN:LIM:STAT OFF` Switches the limit check at the eight corner positions off.

:CALCulate:EGPRs
:RFTX:CORNer[:NORMal]:LIMit:UPPer[:DATA]

Syntax	:CALCulate:EGPRs :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 realx is –150.0, the maximum is 10.0. The minimum resolution possible for realx is 0.01. The default value for realx is 4.0.
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. For further details, please refer to section Corner Points.
Query	There is no query form to this command.
Example	:CALC:EGPR:RFTX:CORN:LIM:UPP 4,4,4,4,4,4,4,4

:CALCulate:EGPRs
:RFTX:CORNer[:NORMal]:LIMit:LOWer[:DATA]

Syntax	:CALCulate:EGPRs :RFTX:CORNer[:NORMal]:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>
Parameters	realx stands for floating point real numbers. The minimum value for realx is –150.0, the maximum is 10.0. The minimum resolution possible for realx is 0.01. The default value for reallx is –150.0.
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. For further details, please refer to section Corner Points.
Query	There is no query form to this command.
Example	:CALC:EGPR:RFTX:CORN:LIM:LOW –150,–150,–150,–150,–150,–150,–150,–150

:CALCulate:EGPRs:RFSPectrum:ACPM:MAVerage?

Syntax	:CALCulate:EGPRs: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:EGPR:ARR:RFSP:ACPM:MOD`
	`:CALC:EGPRs:RFSP:ACPM:MAV?`

`:CALCulate:EGPRs:RFSPectrum:ACPM:MMINimum?`

Syntax	`:CALCulate:EGPRs: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:EGPR:ARR:RFSP:ACPM:MOD`
	`:CALC:EGPRs:RFSP:ACPM:MMIN?`

`:CALCulate:EGPRs:RFSPectrum:ACPM:MMAXimum?`

Syntax	`:CALCulate:EGPRs: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:EGPR:ARR:RFSP:ACPM:MOD`
	`:CALC:EGPRs:RFSP:ACPM:MMAX?`

`:CALCulate:EGPRs:RFSPectrum:ACPM:MSIGma?`

Syntax	`:CALCulate:EGPRs: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, 23 pairs of mean values and standard deviation values are returned.
	The values are floating point real values and separated by commas.

Example	`:CALC:RES`
	`:MEAS:EGPR:ARR:RFSP:ACPM:MOD`
	`...(other SCPI commands)...`
	`:CALC:EGPR:RFSP:ACPM:MSIG?`

`:CALCulate:EGPRs:RFSPectrum:ACPM:MODulation:LIMit`
`[:FAIL]?`

Syntax	`:CALCulate:EGPRs: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 27 boolean numbers in a string, separated by commas. These numbers indicate whether the 27 measurement result values of the `MEAS:EGPRs: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 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:EGPR:BLOC:MSP:AVG 20`
	`:CALC:EGPR:RFSP:ACPM:MOD:LIM?`
	String returned in this example: `"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"`
	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:EGPRs:RFSPectrum:ACPM:MODulation`
`:LIMit:STATe`

Syntax	`:CALCulate:EGPRs:RFSPectrum:ACPM` `:MODulation: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:EGPRs: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:EGPR:RFSP:ACPM:MOD:LIM:STAT ON` Switches the limit check of the ACPM of the spectrum due to switching transients on.

**:CALCulate:EGPRs:RFSPectrum:ACPM:MODulation
:LIMit:UPPer[:DATA]**

Syntax	`:CALCulate:EGPRs: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 available.

Example	`:CALC:EGPR: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:EGPRs:RFSPectrum:ACPM:MODulation:LIMit
:LOWer[:DATA]**

Syntax	`:CALCulate:EGPRs:RFSPectrum:ACPM:MODula-`
	`tion: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 available.
Example	`:CALC:EGPR: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:EGPRs:RFSPectrum:ACPM:TRANsient:LIMit [:FAIL]?`

Syntax	`:CALCulate:EGPRs: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:EGPRs: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, ($\pm$)400, ($\pm$)600, ($\pm$)1.200 and ($\pm$)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:EGPR:ARRay:RFSP:ACPM:TRAN 10`
	`:CALC:EGPR:RFSP:ACPM:TRAN:LIM?`
	String returned in this example: `"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"`
	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:EGPRs:RFSPectrum:ACPM:TRANsient:LIMit :STATe`

Syntax	`:CALCulate:EGPRs: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:EGPRs:RFSP:ACPM:TRAN` measurement against their user-definable limits either `ON` or `OFF`.
Query	The query form of this command is not available.
Example	`:CALC:EGPR:RFSP:ACPM:TRAN:LIM:STAT ON` Switches the limit check of the ACPM of the spectrum due to switching transients on.

:CALCulate:EGPRs:RFSPectrum:ACPM:TRANsient:LIMit :UPPer[:DATA]

Syntax	`:CALCulate:EGPRs: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	There is no query form of this command available.
Example	`:CALC:EGPR:RFSP:ACPM:TRAN:LIM:UPP?` This command will return the 27 upper limit values.

:CALCulate:EGPRs:RFSPectrum:ACPM:TRANsient:LIMit :LOWer[:DATA]

Syntax	`:CALCulate:EGPRs:RFSPectrum:ACPM:TRAN-`
	 `sient: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 sof the ACPM measurement of the spectrum due to switching transients. The physical dimension of the single data values is dBm.

Query	There is no query form of this command available.
Example	`:CALC:EGPR:RFSP:ACPM:TRAN:LIM:LOW?` This command will return the 27 lower limit values.

:CALCulate:EGPRs:RFSPectrum:MSPectrum:VALue

Syntax	`:CALCulate:EGPRs: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:EGPRs:RFSP:MSP:VAL` command is not really a measurement - it is an interpolation of the spectrum curve for the frequency specified.

Example	`:MEAS:EGPR:BLOC:MSP:AVG 20`
	`:CALC:EGPR: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:EGPRs:RFSPectrum:MSPectrum:MAXPeak?`

Syntax	`:CALCulate:EGPRs: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 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:EGPR:RFSP:MSP:MAXP` command is not really a measurement - it is an readout of the spectrum curve.

Example	`:MEAS:EGPR:BLOC:MSP:AVG 20`
	`:CALC:EGPR: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"`.

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 4400 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 4400 in the base channel can be set or altered using the SCPI commands described in section "CONFigure subsystem" on page 426.

– The RF generator and analyzer are enabled using the appropriate `:CONFigure:CSYStem` command.

– For **circuit-switched standard GSM** signals, use of the `:CONFigure:CSYStem G`C**Genana** command.
While working in this mode, a base channel will **not** be provided.

– To set up the 4400 as an RF generator for **both packet-switched and circuit-switched GSM** signals, use the `:CONFigure:CSYStem G`P**Genana** command instead.
In this mode, a base channel can be provided (see command `:RFG:GSM:MOD:CHAN` for details).

:RFGenerator:GPRS:BURSt:TSLot

Syntax	`:RFGenerator:GPRS: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 4400 will transmit its burst. **Notes:** With the help of command `RFG:STYP` you may select whether the Willtek 4400 transmits a continuous wave or bursts. This command is described below. More details regarding timeslots and bursts 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	`:RFGenerator:GPRS:BURSt:TSLot 3` `:RFG:BURS:TSL?` Value returned: `"3"`.

:RFGenerator:GPRS:FREQuency

Syntax	`:RFGenerator:GPRS: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.
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:GPRS:FREQuency 880200010` `:RFG:GPRS:FREQ?` Value returned: `"880200010"`.

:RFGenerator:GPRS:LEVel

Syntax	`:RFGenerator:GPRS: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 4400. 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:GPRS:LEVel -50.5` `:RFG:GPRS:LEV?` Value returned: `"-50.5"`.

:RFGenerator:GPRS:MODulation

Syntax	`:RFGenerator:GPRS: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 4400 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:GPRS:MODulation GMSK` `:RFG:MOD?` Value returned: `"GMSK"`.

:RFGenerator:GPRS:MODulation:AM:MFRequency

Syntax	`:RFGenerator:GPRS: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:GPRS:MOD:AM:MFR 2000` `:RFG:GPRS:MOD:AM:MFR?` Value returned: `"2000"`.

:RFGenerator:GPRS:MODulation:AM:MDEPth

Syntax	`:RFGenerator:GPRS: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:GPRS:MOD:AM:MDEP 50` `:RFG:GPRS:MOD:AM:MDEP?` Value returned: `"50"`.

:RFGenerator:GPRS:MODulation:BITPattern

Syntax	`:RFG:GPRS: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. 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 (`1111111100000000011111111...`).
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:GPRS:MODulation:BITPattern PRBS15` `:RFG:GPRS:MOD:BITP?` Value returned: `"PRBS15"`.

:RFGenerator:GPRS:MODulation:CHANnel

Syntax	`:RFGenerator:GPRS:MODulation:CHANnel <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `GSM｜PDCH｜BPDCh`.
Description	Sets the channel type to be generated. With the `GSM` setting, the Willtek 4400 sends an uncoded bit pattern (with or without training sequence). The `PDCH` setting results in one or multiple time slots with channel encoded Packet Data Channel data. With `BPDCh`, time slot 0 contains FCCH/SCH/BCCH/CCCH, the other slots are filled with PDCH.
Query	The query form of this command returns the current setting. The string delivered back contains one of short-form versions of the predefined expressions explained above.
Example	`:RFG:GPRS:MOD:CHAN BPDCh` `:RFG:GPRS:MOD:CHAN?` Returns the string `BPDC`.

:RFGenerator:GPRS:MODulation:CS

Syntax	`:RFGenerator:GPRS:MODulation:CS <int1>`
Parameters	Valid entries for `int1` are in the range from 1 to 4. The default is 1.
Description	Sets the coding scheme. If `:RFG:GPRS:MOD:CHAN` is set to either `PDCH` or `BPDCh`, the Willtek 4400 transmits a packet data channel which is encoded using this coding scheme.
Query	The query form of this command returns the current setting. The string delivered back contains a one-digit number.
Example	`:RFG:GPRS:MOD:CS 4` `>:RFG:GPRS:MOD:CS?` Returns the following string: 4.

:RFGenerator:GPRS:MODulation:DIFFbitcod

Syntax	`:RFGenerator:GPRS: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:GPRS:MODulation:DIFFbitcod ON` `:RFG:GPRS:MOD:DIFF?` Value returned: `"ON"`.

:RFGenerator:GPRS:MODulation:POLarity

Syntax	:RFGenerator:GPRS: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:GPRS:MODulation:POLarity INVert :RFG:GPRS:MOD:POL? Value returned: "INV".

:RFGenerator:GPRS:MODulation:TSEQuence:STATe

Syntax	:RFGenerator:GPRS: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. **Note:** The type of training sequence can be set using the command 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	:RFG:GPRS:MODulation:TSEQuence:STATe ON :RFG:GPRS:MOD:TSEQ:STAT? Value returned: "ON".

:RFGenerator:GPRS:MODulation:TSEQuence:[DATA]

Syntax	:RFGenerator:GPRS: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.
Query	The query form of this command returns the current setting. A query will return a string, containing one integer.
Example	:RFG:GPRS:MODulation:TSEQuence:DATA 3 :RFG:GPRS:MOD:TSEQ? Value returned: "3".

:RFGenerator:GPRS:MSLOT:LEVel:[DATA]

Syntax	:RFGenerator:GPRS:MSLOT:LEVel:[DATA] \<real1\>

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 Willtek 4400 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:GPRS: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:GPRS:MSLOT:LEVel:MODE

Syntax	`:RFGenerator:GPRS: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 Willtek 4400 Willtek 4400 (as set with the `:RFG: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 explained above.
Example	`:RFG:GPRS:MSLot:LEVel:MODE INDividual` `:RFG:GPRS:MSL:LEV:MODE?` Value returned: `"IND"`.

:RFGenerator:GPRS:STATe

Syntax	`:RFGenerator:GPRS: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 Willtek 4400 Willtek 4400 is in RF generator mode, there is no signaling active. This means that Willtek 4400 may be used as RF signal generator for all kinds of RF and GSM signals. While working as an RF generator, Willtek 4400 Willtek 4400 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:GPRS:STATe ON` `:RFG:GPRS:STAT?` Value returned: `"ON"`.

:RFGenerator:GPRS:STYPe

Syntax	`:RFGenerator:GPRS: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 Willtek 4400 will generate a coded GSM signal **in all** timeslots according to the settings above. When set to `BURSt`, Willtek 4400 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:GPRS:STYPe BURSt` `:RFG:GPRS:STYP?` Value returned: `"BURS"`.

:RFGenerator:EGPRs:BURSt:TSLot

Syntax	`:RFGenerator:EGPRs: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 4400 will transmit its burst. **Notes:** – With the help of command `RFG:STYP` you may select whether the Willtek 4400 transmits a continuous wave or bursts. This command is described below. – More details regarding timeslots and bursts 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	`:RFGenerator:EGPRs:BURSt:TSLot 3`
	`:RFG:BURS:TSL?`
	Value returned in this example `"3"`.

:RFGenerator:EGPRs:FREQuency

Syntax	`:RFGenerator:EGPRs: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:EGPRs:FREQuency 880200010`
	`:RFG:EGPRs:FREQ?`
	Value returned in this example: "880200010".

:RFGenerator:EGPRs:LEVel

Syntax	`:RFGenerator:EGPRs: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 4400. 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:EGPRs:LEVel -50.5`
	`:RFG:EGPRs:LEV?`
	Value returned in this example: "-50.5".

:RFGenerator:EGPRs:MODulation

Syntax	`:RFGenerator:EGPRs: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 (AM). **Notes:** – As soon as GMSK is selected, the Willtek 4400 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 will only available, 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:EGPRs:MODulation GMSK`
	`:RFG:MOD?`
	Value returned in this example "GMSK".

:RFGenerator:EGPRs:MODulation:AM:MFRequency

Syntax	`:RFGenerator:EGPRs: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** that AM will only be available, 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:EGPRs:MOD:AM:MFR 2000`
	`:RFG:EGPRs:MOD:AM:MFR?`
	Value returned in this example: `"2000"`.

:RFGenerator:EGPRs:MODulation:AM:MDEPth

Syntax	`:RFGenerator:EGPRs: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** that AM will only be available, 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:EGPRs:MOD:AM:MDEP 50`
	`:RFG:EGPRs:MOD:AM:MDEP?`
	Value returned in this example: `"50"`.

:RFG:EGPRs:MODulation:BITPattern

Syntax	`:RFG:EGPRs: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 pseudo-random bit sequence. The length of this sequence will be 511 bits ((2 to the power of 9)-1). `PRBS15` is a pseudo-random bit sequence with a length of 32767 bits. `PRBS23` is a pseudo-random 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 (1111111100000000011111111...).
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:EGPRs:MODulation:BITPattern PRBS15`
	`:RFG:EGPRs:MOD:BITP?`
	Value returned in this example: "PRBS15".

:RFGenerator:EGPRs:MODulation:CHANnel

Syntax	`:RFGenerator:EGPRs:MODulation:CHANnel <PredefExp>`
Parameters	`PredefExp` is one of the following predefined expressions: `GSM`|`PDCH`|`BPDCh`.
Description	Sets the channel type to be generated. With the GSM setting, the Willtek 4400 sends an uncoded bit pattern (with or without training sequence). The `PDCH` setting results in one or multiple time slots with channel encoded Packet Data Channel data. With `BPDCh`, time slot 0 contains FCCH/SCH/BCCH/CCCH, the other slots are filled with PDCH.
Query	The query form of this command returns the current setting. The string delivered back contains one of short-form versions of the predefined expressions explained above.

Example	`:RFG:EGPRs:MOD:CHAN BPDCh`
	`:RFG:EGPRs:MOD:CHAN?`
	Returns the string `BPDC`.

:RFGenerator:EGPRs:RFGenerator:EGPRs:MODulation:CS

Syntax	`:RFGenerator:EGPRs:MODulation:CS <int1>`
Parameters	Valid entries for `int1` are in the range from `1` to `4`. The default is `1`.

Description	Sets the coding scheme. If `:RFG:EGPRs:MOD:CHAN` is set to either `PDCH` or `BPDCh`, the Willtek 4400 transmits a packet data channel which is encoded using this coding scheme.
Query	The query form of this command returns the current setting. The string delivered back contains a one-digit number.

Example	`:RFG:EGPRs:MOD:CS 4`
	`>:RFG:EGPRs:MOD:CS?`
	Returns the following string: 4.

:RFGenerator:EGPRs:MODulation:DIFFbitcod

Syntax	`:RFGenerator:EGPRs: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:EGPRs:MODulation:DIFFbitcod ON`
	`:RFG:EGPRs:MOD:DIFF?`
	Value returned in this example "`ON`".

:RFGenerator:EGPRs:MODulation:POLarity

Syntax	`:RFGenerator:EGPRs: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:EGPRs:MODulation:POLarity INVert`
	`:RFG:EGPRs:MOD:POL?`
	Value returned in this example "`INV`".

:RFGenerator:EGPRs:MODulation:TSEQuence:STATe

Syntax	`:RFGenerator:EGPRs: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:EGPRs:MODulation:TSEQuence:STATe ON`
	`:RFG:EGPRs:MOD:TSEQ:STAT?`
	Value returned in this example "ON".

:RFGenerator:EGPRs:MODulation:TSEQuence:[DATA]

Syntax	`:RFGenerator:EGPRs: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:EGPRs:MODulation:TSEQuence:DATA 3`
	`:RFG:EGPRs:MOD:TSEQ?`
	Value returned in this example "3".

:RFGenerator:EGPRs:MSLOT:LEVel:[DATA]

Syntax	`:RFGenerator:EGPRs:MSLOT:LEVel:[DATA] <real1>`
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 Willtek 4400 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:EGPRs:MSLot:LEVel:DATA -80.0,-70.0,-60.0,-50.0,-40.0, -30.0,-20.0,-10.0`
	`:RFG:MSL:LEV?`
	Values returned in this example: "-80.0,-70.0,-60.0,-50.0,-40.0, -30.0,-20.0,-10.0"

:RFGenerator:EGPRs:MSLOT:LEVel:MODE

Syntax	`:RFGenerator:EGPRs: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 Willtek 4400 Willtek 4400 (as set with the `:RFG:LEV` command) will be used. **Note:** For a detailed explanation of the individual power levels in conjunction with the multislot measurement capabilities of Willtek 4400 (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:EGPRs:MSLot:LEVel:MODE INDividual` `:RFG:EGPRs:MSL:LEV:MODE?` Value returned in this example: `"IND"`.

:RFGenerator:EGPRs:STATe

Syntax	`:RFGenerator:EGPRs: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 Willtek 4400 Willtek 4400 is in RF generator mode, there is no signalling active. This means that Willtek 4400 may be used as RF signal generator for all kinds of RF and GSM signals. While working as a RF generator, Willtek 4400 Willtek 4400 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:EGPRs:STATe ON` `:RFG:EGPRs:STAT?` Value returned in this example "`ON`".

:RFGenerator:EGPRs:STYPe

Syntax	`:RFGenerator:EGPRs: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 Willtek 4400 will generate a coded GSM signal in all timeslots according to the settings above. When set to `BURSt`, Willtek 4400 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:EGPRs:STYPe BURSt`
	`:RFG:EGPRs:STYP?`
	Value returned in this example `"BURSt"`.

RFANalyser subsystem

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

:RFANalyser:GPRS:FREQuency

Syntax	:RFANalyser:GPRS: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. **Note:** The frequency of the Willtek 4400's 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	:RFANalyser:GPRS:FREQuency 880200000 :RFAN:GPRS:FREQ? Value returned: "880200000".

:RFANalyser:GPRS:MODulation

Syntax	:RFANalyser:GPRS: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:GPRS:MODulation GMSK ::RFANalyser:GPRS:MODulation? Value returned: "GMSK".

RFSPectrum subsystem

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

:RFSPectrum[:GSM│GPRS│EGPRs]:MSPectrum:SPAN

Syntax	:RFSPectrum[:GSM│GPRS│EGPRs]: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|GPRS|EGPRs]:MSPectrum:RESolution

Syntax	:RFSPectrum[:GSM|GPRS|EGPRs]: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 4400 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".

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 4400.

: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 4400.
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 4400. `CODec` means that the signal created by the AF generator will be transmitted over the traffic channel (GSM and AMPS only). **Notes** – 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. – The codec is an option to the Willtek 4400. Please refer to section "Accessories and options" on page 29 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: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 4400. 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: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 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 4400 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 4400. **AFIN** means that the signal applied to the Willtek 4400 **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 4400. **Notes** – The audio generator and the codec are options to the Willtek 4400. For further details, please refer to section "Accessories and options" on page 29. – For further details regarding the connectors of the Willtek 4400, please refer to section "Connectors" on page 9. – 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 4400 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 4400.

:AFANalyser:INPut

Syntax	`:AFANalyser:INPut <PredefExp>`
Parameters	**PredefExp** is one of the following predefined expressions: **AF|AUXin**. Default is **AF**.
Description	Selects the input signal for the AF analyzer of the Willtek 4400. **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 4400 (RF section). **Notes** – This command requires the Audio Option to be installed on your Willtek 4400. – The codec is an option to the Willtek 4400. For details, please refer to section "Accessories and options" on page 29. – Input voltage limitations apply on the AF in and the AUX in connectors of the Willtek 4400. Please refer to section "Connectors" on page 9 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	`:AFANalyser:INPut AUXin` `:AFAN:INP?` Value returned: **"AUX"**.

: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 4400.
	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 4400 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 4400 "AF in" connector. AUTO means that the Willtek 4400 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 4400 "AUX 4 (AUX in)" connector. **AUTO** means that the Willtek 4400 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 4400 either on or off. Please note that this command needs the power supply option to be installed on your Willtek 4400. Details regarding this option can be found in section "Accessories and options" on page 29.
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 4400. Details regarding this option can be found in section "Accessories and options" on page 29.
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 4400. Details regarding these options can be found in section "Accessories and options" on page 29.
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"`.

Warranty and Repair

<div style="text-align: right;">

<table><tr><td>B</td></tr></table>

</div>

This chapter describes the customer services available through Willtek. Topics discussed in this chapter include the following:

–
–

Warranty information

Willtek warrants that all of its products conform to Willtek's published specifications and are free from defects in materials and workmanship for a period of one year from the date of delivery to the original buyer, when used under normal operating conditions and within the service conditions for which they were designed. This warranty is not transferable and does not apply to used or demonstration products.

In case of a warranty claim, Willtek's obligation shall be limited to repairing, or at its option, replacing without charge, any assembly or component (except batteries) which in Willtek's sole opinion proves to be defective within the scope of the warranty. In the event Willtek is not able to modify, repair or replace nonconforming defective parts or components to a condition as warranted within a reasonable time after receipt thereof, the buyer shall receive credit in the amount of the original invoiced price of the product.

It is the buyer's responsibility to notify Willtek in writing of the defect or nonconformity within the warranty period and to return the affected product to Willtek's factory, designated service provider, or authorized service center within thirty (30) days after discovery of such defect or nonconformity. The buyer shall prepay shipping charges and insurance for products returned to Willtek or its designated service provider for warranty service. Willtek or its designated service provider shall pay costs for return of products to the buyer.

Willtek's obligation and the customer's sole remedy under this hardware warranty is limited to the repair or replacement, at Willtek's option, of the defective product. Willtek shall have no obligation to remedy any such defect if it can be shown: (a) that the product was altered, repaired, or reworked by any party other than Willtek without Willtek's written consent; (b) that such defects were the result of customer's improper storage, mishandling, abuse, or misuse of the product; (c) that such defects were the result of customer's use of the product in conjunction with equipment electronically or mechanically incompatible or of an inferior quality; or (d) that the defect was the result of damage by fire, explosion, power failure, or any act of nature.

The warranty described above is the buyer's sole and exclusive remedy and no other warranty, whether written or oral, expressed or implied by statute or course of dealing shall apply. Willtek specifically disclaims the implied warranties of merchantability and fitness for a particular purpose. No statement, representation, agreement, or understanding, oral or written, made by an agent, distributor, or employee of Willtek, which is not contained in the foregoing warranty will be binding upon Willtek, unless made in writing and executed by an authorized representative of Willtek. Under no circumstances shall Willtek be liable for any direct, indirect, special, incidental, or consequential damages, expenses, or losses, including loss of profits, based on contract, tort, or any other legal theory.

Equipment return instructions

Please contact your local service center for Willtek products via telephone or web site for return or reference authorization to accompany your equipment. For each piece of equipment returned for repair, attach a tag that includes the following information:

– Owner's name, address, and telephone number.

– Serial number, product type, and model.

– Warranty status. (If you are unsure of the warranty status of your instrument, include a copy of the invoice or delivery note.)

– Detailed description of the problem or service requested.

– Name and telephone number of the person to contact regarding questions about the repair.

– Return authorization (RA) number or reference number.

If possible, return the equipment using the original shipping container and material. Additional Willtek shipping containers are available from Willtek on request. If the original container is not available, the unit should be carefully packed so that it will not be damaged in transit. Willtek is not liable for any damage that may occur during shipping. The customer should clearly mark the Willtek-issued RA or reference number on the outside of the package and ship it prepaid and insured to Willtek.

Publication History

Revision	Comment
0112-200-A	First edition
0112-201-A	No changes (manual for bug fix release 2.01)
0203-202-A	No changes (manual for bug fix release 2.02)
0207-310-A	Enhanced GSM and GPRS features (including multislot).
0301-320-A	MIN-MAX graphical display function, TCP/IP functions, IMEI-SV display with check digit, coupling loss files in subdirectories.
0310-400-A	CDMA2000 1xRTT added to 4400 functionality.
0420-420-A	EDGE System Option added.
0411-500-A	New software version 5.0 that includes all different system options for the 4400 series. EDGE SCPI commands updated.
0506-510-A	New software version 5.10 including improved EGPRS attach procedure to allow for all phones to register. Performed some corrections in the RAPID chapter.
0508-610-A	New software version 6.10. Screen Shots updated. SCPI commands added.
0602-620-A	New software versions 6.20 and 11.20.

will'tek