

4400 Mobile Phone Tester Series 4464 CDMA2000 System Option



User's guide
Version 6.20

boosting wireless efficiency

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

Copyright © Copyright 2006 Willtek Communications GmbH. All rights reserved. Willtek and its logo are trademarks of Willtek Communications. All other trademarks and registered trademarks are the property of their respective owners. No part of this guide may be reproduced or transmitted electronically or otherwise without written permission of the publisher.

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

Specifications, terms and conditions are subject to change without notice. All trademarks and registered trademarks are the property of their respective companies.

Ordering information This guide is issued as part of the **Willtek 4464 CDMA2000 System Option**. The ordering number for a published guide is M 292 010. 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 4464 CDMA2000 System Option | M 248 663 |

Compliance statements This manual refers to the CDMA2000 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.

Table of Contents

About This Guide

| | |
|---|-----|
| Index of the 4463 GSM/GPRS System Option, 4468 EDGE System Option . . . | xiv |
| Purpose and scope | xiv |
| Assumptions | xiv |
| Related information | xv |
| Technical assistance | xv |
| Conventions | xv |

Safety Notes

| | |
|---|-------|
| Safety class | xvii |
| Before startup | xviii |
| During test | xviii |
| During maintenance and repair | xviii |
| Shutdown when defective | xix |

Chapter 1

| | |
|---|----------|
| Overview | 1 |
| About the 4400 | 2 |
| What's new in version 6.20 | 3 |
| Features and capabilities | 3 |
| Connecting test leads | 4 |
| Crash course | 4 |
| Cabling | 4 |
| Cable connection | 5 |
| Air connection | 5 |
| Using the front and rear panels | 6 |
| Front panel | 7 |
| Rear panel | 8 |
| Connectors | 9 |
| Connectors on the front panel | 9 |
| Connectors on the rear panel | 10 |

| | |
|--|----|
| The keys on the front panel..... | 14 |
| The softkeys..... | 15 |
| The marker tabs..... | 16 |
| The function keys..... | 16 |
| The numeric keys..... | 18 |
| The cursor keys..... | 18 |
| The selection key..... | 18 |
| Keyboard mapping table..... | 19 |
| Menus and levels..... | 20 |
| The areas of a test menu..... | 21 |
| Menu fields..... | 22 |
| Entry fields..... | 22 |
| Display fields..... | 25 |
| Help on Help..... | 26 |
| Navigating help..... | 27 |
| Using the online help on an external PC..... | 28 |
| Installing the online help on a PC..... | 28 |
| Navigating through the help pages..... | 28 |
| Accessories and options..... | 29 |
| Expanding the 4400's measurement capabilities..... | 29 |
| Little aids for speeding up tests..... | 31 |
| Rack installation, cases..... | 35 |

Chapter 2

| | |
|---|-----------|
| CDMA Call Mode | 37 |
| Overview..... | 38 |
| Common elements of the call mode menus..... | 38 |
| Test parameters..... | 38 |
| Status..... | 40 |
| Display markers..... | 41 |
| Statistical test evaluation..... | 41 |
| Softkeys in idle mode..... | 42 |
| MS Call..... | 43 |
| BS Call..... | 43 |
| Softkeys at call level..... | 43 |
| Testing the phone on CDMA channels..... | 45 |
| Testing voice or data?..... | 45 |
| Basic test menu..... | 46 |
| Call graph..... | 47 |
| Basic test results display..... | 49 |
| Other screen elements..... | 51 |
| Rho/Frequency test menu..... | 51 |
| Overview of power tests..... | 53 |
| Min./max. power test..... | 54 |
| Open loop power control test menu..... | 56 |
| Gated power test menu..... | 57 |
| Closed loop power control menu..... | 60 |
| Access probe menu..... | 62 |
| FER test menu..... | 63 |
| Reported FER (voice service)..... | 64 |
| FER (loopback service)..... | 66 |
| Code Domain test menu..... | 68 |
| MS Info test menu..... | 70 |

| | |
|--|-----|
| Hard Handoff menu..... | 71 |
| System parameters | 73 |
| Forward link | 74 |
| Sector 1 setup..... | 74 |
| QPCH..... | 75 |
| Sector 2 setup..... | 75 |
| Channel setup..... | 76 |
| AWGN..... | 76 |
| Network setup | 77 |
| Network information..... | 77 |
| Access channel parameters..... | 78 |
| Traffic channel | 79 |
| Common parameters..... | 79 |
| Fundamental channel parameters | 80 |
| Supplemental channel parameters | 81 |
| Limits menus..... | 83 |
| How to read and enter the measurement limits..... | 83 |
| Limits..... | 84 |
| Limits continued..... | 86 |
| Gated PTT..... | 87 |
| Open loop..... | 88 |
| Coupling loss..... | 89 |
| How to activate a previously stored coupling loss definition | 90 |
| How to store a coupling loss definition on the 4400 | 91 |
| How to edit a coupling loss definition already stored on the 4400 .. | 92 |
| Entering coupling loss values from the Manual Input menu..... | 92 |
| Auxiliary outputs | 94 |
| CDMA basics | 95 |
| Basic specifications of CDMA bands..... | 95 |
| Radio configurations, service options..... | 97 |
| Test modes..... | 99 |
| Voice testing | 99 |
| Loopback testing | 99 |
| Data channel testing..... | 100 |
| SCM coding..... | 100 |
| MS power classes..... | 101 |

Chapter 3

| | |
|--|------------|
| CDMA Nonsignaling Mode | 103 |
| Overview | 104 |
| RF generator | 105 |
| RF analyzer for rho and frequency | 107 |
| RF analyzer for power..... | 107 |
| RF analyzer for gated power..... | 109 |
| Function..... | 109 |
| How to read the gated power test results display..... | 109 |
| Viewing certain details of the gated power | 109 |
| Softkeys in the gated power menu | 110 |
| RF analyzer for code domain power | 110 |
| Function..... | 111 |
| How to read the code domain test results display | 111 |
| Code domain power parameters | 111 |
| Common elements of the RF analyzer menus..... | 112 |

| | |
|---------------------------------------|-----|
| Entry fields of the RF analyzer | 112 |
| Softkeys of the RF analyzer | 113 |
| Parameter menus | 113 |

| | | |
|------------------|--|------------|
| Chapter 4 | AMPS Call Mode | 115 |
| | Overview | 116 |
| | Common elements of the call mode menus | 117 |
| | Test parameters | 117 |
| | Status | 118 |
| | Statistical test evaluation | 119 |
| | Softkeys | 120 |
| | Testing the phone on AMPS channels | 120 |
| | TX Tests menu | 121 |
| | What it does | 121 |
| | Results display | 121 |
| | Selecting measurement-specific parameters | 122 |
| | RX Tests menu | 123 |
| | What it does | 123 |
| | How to read the test results display | 124 |
| | ST Tests | 124 |
| | What it does | 125 |
| | How to read the results display | 125 |
| | Selecting measurement-specific parameters | 125 |
| | Parameters | 125 |
| | How to read and enter the measurement limits | 126 |
| | Limits | 126 |

| | | |
|------------------|--|------------|
| Chapter 5 | AMPS Nonsignaling Mode | 129 |
| | Overview | 130 |
| | RF generator | 130 |
| | Transmitter tests | 132 |
| | Receiver tests | 133 |
| | Signaling tone tests | 134 |
| | Common elements of the RF analyzer menus | 135 |
| | Entry fields of the RF analyzer | 135 |
| | Softkeys of the RF analyzer | 135 |
| | Parameter menus | 136 |

| | | |
|------------------|---|------------|
| Chapter 6 | Tools | 137 |
| | Overview | 138 |
| | Configuration | 138 |
| | Access to the configuration menus | 138 |
| | Setup | 139 |
| | Settings of the Setup menu | 139 |
| | I/O configuration | 140 |
| | IEEE 488.2 port settings | 140 |
| | TCP/IP port settings | 141 |
| | TCP/IP Troubleshooting | 143 |
| | Parallel port settings | 144 |

| | |
|--|-----|
| Options..... | 145 |
| Basic info area | 145 |
| Options..... | 146 |
| Installing additional options | 147 |
| Service and software update..... | 148 |
| Changing the version of the system software..... | 148 |
| The softkeys of this menu | 149 |
| Initiating a software update from a remote PC..... | 150 |
| Utilities..... | 151 |
| Access to the Utilities menus | 151 |
| I/O trace for GPIB communications..... | 151 |
| The softkeys of this menu | 152 |
| Info trace..... | 152 |
| The softkeys of this menu | 153 |
| MS power supply | 153 |
| MS Power Supply Option | 154 |
| Current Measurement Option | 155 |
| Spectrum measurements..... | 158 |
| Access to the spectrum measurement menus..... | 158 |
| Modulation spectrum measurements..... | 158 |
| On first sight..... | 159 |
| The spectrum live display..... | 159 |
| ACPM modulation spectrum measurements..... | 161 |
| On first sight..... | 162 |
| Understanding the measurement principle | 162 |
| The spectrum live display..... | 163 |
| Softkeys | 165 |
| Display markers | 166 |
| Audio measurements | 166 |
| Audio measurements – test basics | 166 |
| Input and output ports for audio measurements..... | 166 |
| Access to the audio menus | 168 |
| Audio frequency generator | 168 |
| Using the AF generator..... | 168 |
| Wave file replay | 170 |
| AF generator menu – softkeys | 172 |
| AF generator – parameters | 173 |
| AF basic analyzer..... | 173 |
| The AF basic analyzer menu | 173 |
| Current measurement results | 174 |
| AF analyzer – softkeys | 175 |
| AF analyzer – parameters..... | 176 |
| Statistical data evaluation..... | 177 |
| Parameters area | 179 |
| AF spectrum analyzer..... | 180 |
| The AF spectrum analyzer menu | 180 |

Chapter 7

| | |
|-------------------|------------|
| RAPID! | 183 |
| Overview | 184 |
| Using RAPID!..... | 184 |

| | |
|---|-----|
| Introduction | 185 |
| RAPID! = BASIC + SCPI | 185 |
| Entering and exiting the RAPID! environment | 185 |
| Marker tabs of the RAPID! environment | 186 |
| File menu | 187 |
| File name – directory – selection area | 187 |
| Browser area | 188 |
| RAPID! basic file area | 189 |
| Softkeys of the file menu | 189 |
| Edit menu | 191 |
| Edit area | 192 |
| Status area | 192 |
| Softkeys of the Edit menu | 193 |
| Search/Replace menu | 193 |
| Typing tabs | 194 |
| Run menu | 195 |
| Output area | 196 |
| Softkeys of the Run menu | 197 |
| Debug menu | 197 |
| Source area | 198 |
| Variable area | 198 |
| Softkeys of the Debug menu | 200 |
| RAPID! syntax | 201 |
| General syntax | 201 |
| BASIC form | 201 |
| Program and line formats | 201 |
| Basic rules | 202 |
| Syntax check | 202 |
| Notation | 202 |
| Program lines | 203 |
| Variables | 204 |
| Constants | 205 |
| Operators | 206 |
| Expressions | 207 |
| Numeric expressions | 208 |
| String expressions | 208 |
| Boolean expressions | 208 |
| Commands | 209 |
| Overview | 209 |
| General commands | 210 |
| CHAIN | 210 |
| END | 211 |
| LET | 211 |
| REM or ' (Comments) | 211 |
| STOP | 212 |
| WAIT | 212 |
| Screen commands | 212 |
| CLS | 213 |
| INPUT | 214 |
| LOCATE | 214 |
| PRINT (OUTPUT) | 215 |
| SOFTKEYS | 216 |
| TEXTATTR | 216 |

| | |
|---|-----|
| Commands related to variables, procedures and functions | 217 |
| DIM | 217 |
| ERASE | 217 |
| FUNCTION – EXIT FUNCTION – END FUNCTION | 218 |
| GLOBAL | 219 |
| SUB – EXIT SUB – END SUB – CALL | 219 |
| VARIABLE | 220 |
| Control commands | 221 |
| DO ... LOOP, WHILE, UNTIL | 221 |
| FOR ... NEXT | 222 |
| IF ... THEN, ELSE, ELSEIF, END IF | 223 |
| SELECT CASE, CASE ELSE, END SELECT | 224 |
| Branch commands | 225 |
| ERROR | 225 |
| GOSUB ... RETURN | 225 |
| GOTO | 226 |
| ON ERROR, RESUME | 226 |
| Commands for input/output handling | 228 |
| CLOSE | 228 |
| INPUT | 228 |
| OPEN (on files) | 229 |
| OPEN (on communication ports) | 230 |
| PRINT or OUTPUT | 233 |
| Functions | 234 |
| Numeric functions | 235 |
| BIN | 235 |
| BIN\$ | 235 |
| CINT | 236 |
| HEX | 236 |
| HEX\$ | 236 |
| OCT | 237 |
| OCT\$ | 237 |
| VAL | 237 |
| VAL\$ | 238 |
| String functions | 238 |
| ASC | 238 |
| CHR\$ | 239 |
| INSTR | 239 |
| LEN | 239 |
| LEFT\$ | 240 |
| MID\$ | 240 |
| RIGHT\$ | 240 |
| SPACE\$ | 241 |

| | |
|-------------------------------------|-----|
| I/O functions..... | 241 |
| CHDIR..... | 242 |
| CLOCK..... | 242 |
| CURDIR\$ | 242 |
| DATE\$..... | 242 |
| DIR\$ | 243 |
| EOF | 243 |
| EVENTWAIT, EVENTSTATUS | 244 |
| FREEFILE..... | 245 |
| INKEY | 245 |
| INKEYWAIT | 246 |
| KILL | 246 |
| MKDIR..... | 246 |
| NAME | 247 |
| RMDIR..... | 247 |
| SHELL | 247 |
| TIME\$ | 248 |
| Functions for error handling..... | 248 |
| ERR | 248 |
| ERL..... | 249 |
| ERF\$ | 249 |
| Mathematical functions..... | 249 |
| ABS | 250 |
| LOG, LGT..... | 250 |
| RND, RANDOMIZE | 250 |
| SIN, COS, TAN, ATAN..... | 251 |
| SQRT | 251 |
| SGN..... | 251 |
| Tables..... | 252 |
| RAPID! commands and functions | 252 |
| Syntax errors..... | 255 |
| Runtime errors..... | 259 |

Chapter 8

| | |
|---------------------------------|------------|
| SCPI | 263 |
| Overview..... | 264 |
| What SCPI is..... | 264 |
| Structure..... | 265 |
| Syntax and notation..... | 265 |
| Compound commands | 266 |
| Parameters..... | 267 |
| Queries..... | 267 |
| Common commands..... | 267 |
| SCPI notation | 268 |
| SCPI and RAPID!..... | 269 |
| Executing SCPI commands | 269 |
| Example 1 | 269 |
| Example 2 | 269 |
| Example 3 | 270 |
| Reading SCPI data..... | 270 |
| Using queries | 271 |
| Building queries..... | 271 |
| Event handling – registers..... | 272 |

| | |
|---|-----|
| Programming examples | 272 |
| Standard TX measurements..... | 273 |
| Message exchange..... | 274 |
| Example of a GPIB protocol..... | 275 |
| Command subsystem overview | 276 |
| Using the SCPI commands..... | 276 |
| Schematic view of the subsystems of the 4400..... | 276 |
| Common commands..... | 277 |
| The communication-related subsystems | 278 |
| The SYSTem subsystem..... | 278 |
| The STATus subsystem..... | 278 |
| Understanding the STATus subsystem..... | 279 |
| Table of registers | 282 |
| The PROGRAM subsystem - overview..... | 285 |
| The FORMat subsystem - overview | 285 |
| The Node B and UE parameter subsystems | 286 |
| The CONFIgure subsystem | 286 |
| The CALL subsystem | 287 |
| The Measurement subsystems | 287 |
| The MEASure subsystem | 287 |
| :MEASure[:CONTInuous]..... | 289 |
| MEASure:ARRay | 290 |
| :MEAS[:CONT]:BLOCkdata | 292 |
| :MEAS:BLOC:MSP[:CURRent] | 293 |
| :MEASure:::GROup | 294 |
| The FETCh subsystem | 295 |
| FETCh:LAST..... | 295 |
| FETCh:BLOCkdata:::?. | 296 |
| FETCh<{:measProp}>..... | 296 |
| The CALCulate Subsystem | 297 |
| Reading the basic scheme | 298 |
| CALCulate:LIMit | 299 |
| CALCulate:{Statistics} | 301 |
| Using limits | 301 |
| Working with complex limits | 303 |
| Measurement device configuration subsystems | 304 |
| RF measurement devices | 304 |
| AF measurement devices | 304 |
| The RFGenerator subsystem | 304 |
| The RFANalyser subsystem | 305 |
| The RFSpectrum subsystem..... | 305 |
| The AFGenerator subsystem | 305 |
| The AFANalyser subsystem | 305 |
| The MS Power Supply subsystem | 305 |
| SCPI command errors..... | 305 |

Appendix A

| | |
|---|------------|
| SCPI Command Reference | 309 |
| Signaling operation register group | 310 |
| Common commands..... | 311 |
| General common commands..... | 311 |
| Commands affecting the event status register..... | 312 |
| Commands affecting the service register..... | 313 |

| | |
|---------------------------------|-----|
| SYSTem subsystem | 316 |
| STATus subsystem | 324 |
| PROGram subsystem..... | 340 |
| FORMat subsystem | 343 |
| CONFigure subsystem..... | 345 |
| CALL subsystem | 369 |
| MEASure subsystem | 377 |
| FETCh Subsystem..... | 417 |
| CALCulate Subsystem..... | 438 |
| READ subsystem | 503 |
| RFGenerator subsystem | 503 |
| RFANalyser subsystem | 516 |
| RFSpectrum subsystem | 517 |
| AFGenerator subsystem..... | 519 |
| AFANalyser subsystem | 523 |
| MS Power Supply subsystem | 527 |

| | | |
|-------------------|------------------------------------|------------|
| Appendix B | Warranty and Repair | 529 |
| | Warranty information..... | 530 |
| | Equipment return instructions..... | 531 |

| | |
|----------------------------|------------|
| Publication History | 533 |
|----------------------------|------------|

About This Guide

- ["Index of the 4463 GSM/GPRS System Option, 4468 EDGE System Option"](#) on page xiv
- ["Purpose and scope"](#) on page xiv
- ["Assumptions"](#) on page xiv
- ["Related information"](#) on page xv
- ["Technical assistance"](#) on page xv
- ["Conventions"](#) on page xv

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

["WCDMA Non-Call Mode"](#) – Testing a mobile phone in service mode

["WCDMA Call Mode"](#) – Description of the tests and systems which can be tested using call setup

["Tools"](#) – Description of the available tools

["The 4484 WCDMA Tracer Option"](#) – Description of the optional tracer software tool

["RAPID!"](#) – Introduction, syntax, language elements, error messages

["SCPI"](#) – General introduction and structure

SCPI Command Reference – The commands for the 4464 CDMA2000 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 4464 CDMA2000 System Option features and capabilities. This guide includes task-based instructions that describe how to install, configure, use, and troubleshoot the 4464 CDMA2000 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 4464 CDMA2000 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 <code>typeface</code> . | All results okay |
| Text you must type exactly as shown appears in this <code>typeface</code> . | Type: <code>a : \set.exe</code> in the dialog box. |
| Variables appear in this <code><typeface></code> . | Type the new <code><hostname></code> . |
| Book references appear in this <code>typeface</code> . | Refer to <code>Newton's Telecom Dictionary</code> |
| A vertical bar means "or": only one option can appear in a single command. | <code>platform [a b e]</code> |

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 xviii
- "Before startup" on page xviii
- "During test" on page xviii
- "During maintenance and repair" on page xviii
- "Shutdown when defective" on page xix

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 \pm 20% or 240 V \pm 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" on page 2](#)
- ["What's new in version 6.20" on page 3](#)
- ["Features and capabilities" on page 3](#)
- ["Connecting test leads" on page 4](#)
- ["Using the front and rear panels" on page 6](#)
- ["Menus and levels" on page 20](#)
- ["Menu fields" on page 22](#)
- ["Help on Help" on page 26](#)
- ["Accessories and options" on page 29](#)

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 Spectrum 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-SCDMA 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 order 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

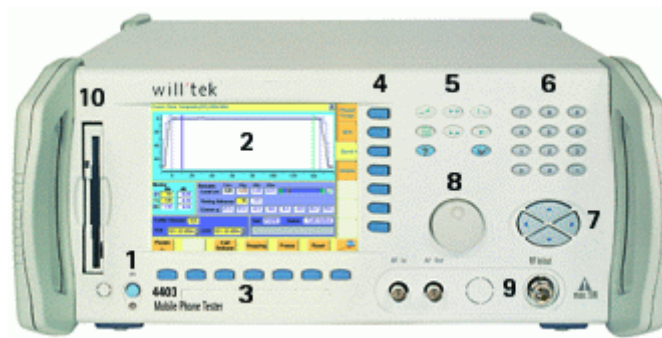
[Front panel](#) – Get to know the ten sections of the front panel.

[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](#).



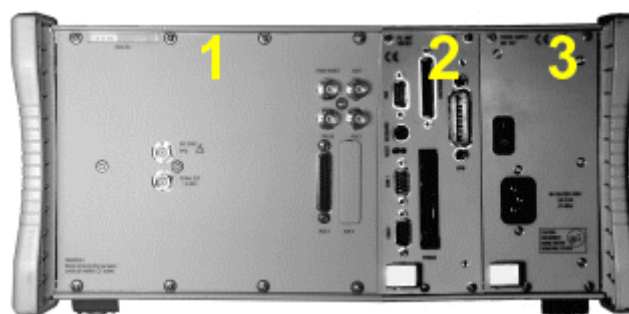
HAZARD

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 k Ω /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 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.

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.



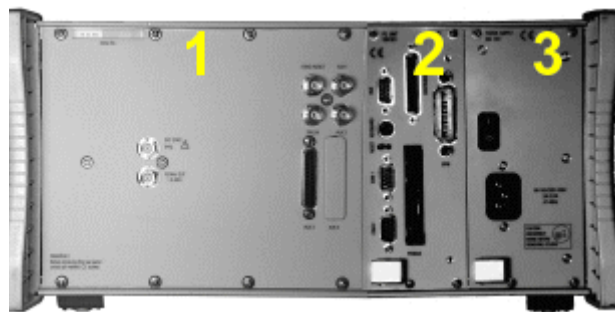
HAZARD

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.

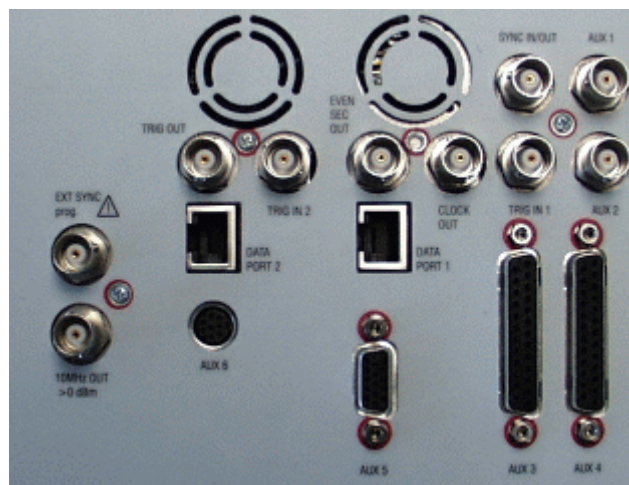
Note: There is no time limitation for applying the maximum input power.

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 k Ω .

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 cannot 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.

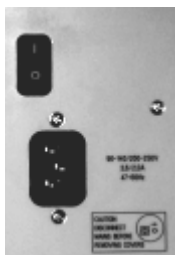
CENTRONICS — Standard parallel port to connect a PC printer (LPT 1).

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 — General Purpose Interface Bus according to specification IEEE-488.2.

Connector on the power supply



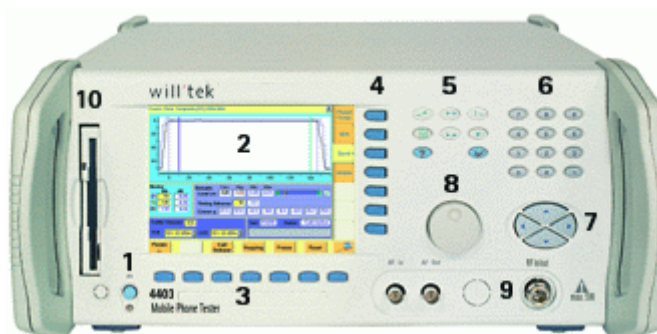
Power in — Standard IEC 320 power connector (3 pins). Please make sure that the power cable is properly grounded.



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.

Param

...

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



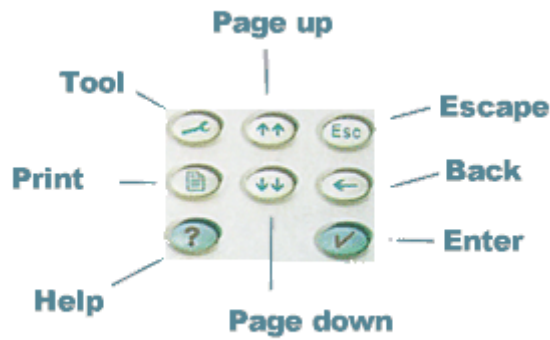
Marker tab [MS Info], blocked

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).

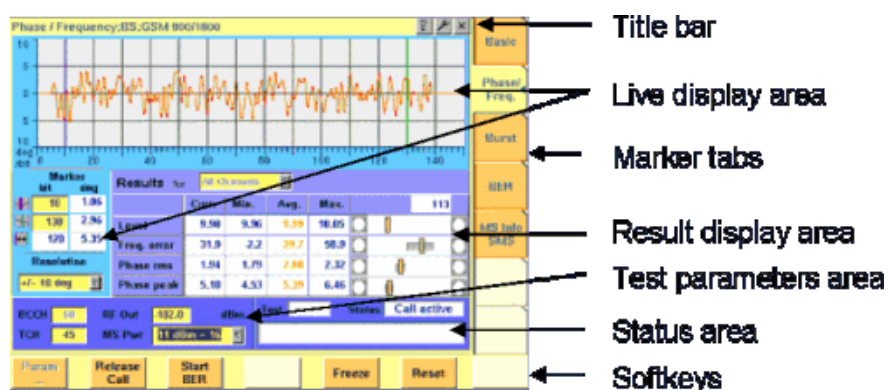
Now you push the third marker tab (from the top) and the 4400 will display the Burst test menu.

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 characteristics 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 alphanumeric (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.



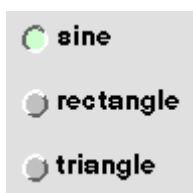
Example: **TCH Type** – This selection field enables you to choose a specific TCH type from a list.

- **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

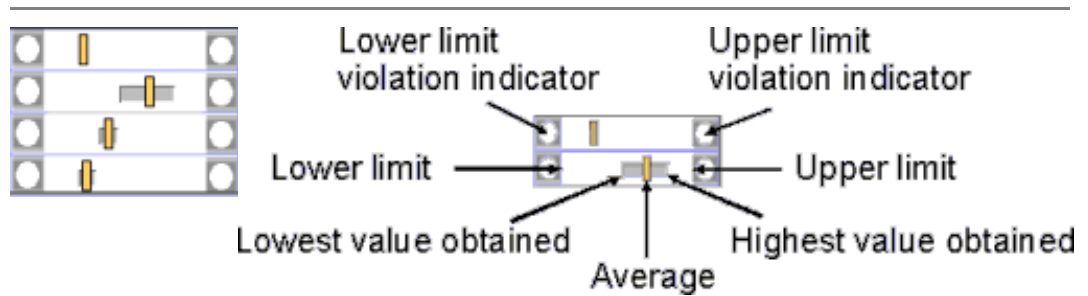
The 4400 offers you two different types of display fields:

(Alpha)numeric display fields

| MS Report | | |
|-----------|----------------|---------|
| | RX Lev | RX Qual |
| Full | 63 = -48.0 dBm | 0 |
| Sub | 63 = -48.0 dBm | 0 |

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

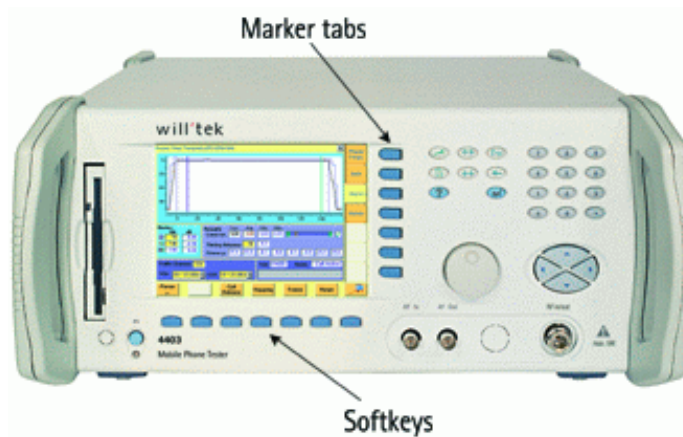
While working with the Online help, you will have to push softkeys, marker tabs, or other keys. To keep things as clear as possible, the online manual either uses the graphical image of the item as you see it on-screen or it uses the following notation rules:

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.

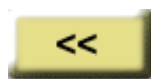
Example: This page provides links to the documents explaining the softkeys and marker tabs in detail. Use the marker tabs to directly jump to the related pages.

Softkeys – the quick navigators

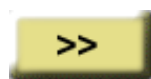
Using the softkeys you quickly navigate through a chapter or call its index. There are five softkeys available:



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

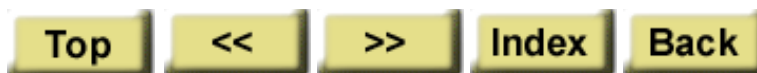
Links instead of marker tabs

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 losing 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 ACPM 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 187 and "[Commands for input/output handling](#)" on page 228).
- 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 background 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 |

CDMA Call Mode

2

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

- ["Overview" on page 38](#)
- ["Common elements of the call mode menus" on page 38](#)
- ["Testing the phone on CDMA channels" on page 45](#)
- ["System parameters" on page 73](#)
- ["CDMA basics" on page 95](#)

Overview

This section covers information on cdmaOne and CDMA2000 measurements in synchronous (call) mode. For information on CDMA measurements, see ["Testing the phone on CDMA channels" on page 45](#). See also ["Common elements of the call mode menus" on page 38](#) for more information. Additional test parameters and limits are explained in section ["System parameters" on page 73](#), and section ["CDMA basics" on page 95](#) covers some more general aspects of the CDMA standards.

To select the synchronous test mode, press the **Calls & Meas.** softkey on the Welcome menu.

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

Table 1 CDMA frequency bands supported by the 4400

| Band description | Reverse link (MS transmit) | Forward link (BS transmit) |
|--------------------|----------------------------|----------------------------|
| U.S. cellular | 824 to 849 MHz | 869 to 894 MHz |
| North American PCS | 1850 to 1910 MHz | 1930 to 1990 MHz |
| TACS | 872 to 915 MHz | 917 to 960 MHz |
| JTACS | 887 to 925 MHz | 832 to 870 MHz |
| Korean PCS | 1850 to 1780 MHz | 1840 to 1870 MHz |
| NMT-450 | 411 to 484 MHz | 421 to 494 MHz |
| IMT-2000 | 1920 to 1980 MHz | 2110 to 2170 MHz |
| 900 MHz | 880 to 915 MHz | 925 to 960 MHz |
| 1800 MHz | 1710 to 1785 MHz | 1805 to 1880 MHz |

Common elements of the call mode menus

Test parameters

The screenshot shows a menu with the following fields and values:

- CCH: 384
- RF Out: -70.0 dBm
- TCH: 29
- RC: F-RC3 & R-RC3
- Band: US 800
- SO: 2
- SID: 80

These entry and display fields offer the possibility to easily view and in some cases, alter important test parameters.

CCH — Channel number for the control channel. This is the frequency channel at which the 4400 transmits the logical channels that the mobile requires to synchronize with the simulated network and to set up a call. The valid input range depends on the selected frequency band.

NOTE

Section ["Basic specifications of CDMA bands" on page 95](#) gives an overview of the channel numbers used in the various CDMA bands. With the help of the information provided there, it is also possible to calculate the RF frequency from the channel number.

RF Out — RF output level of the 4400's signal, or the input level at the mobile phone if the coupling loss is selected correctly. The valid range is from -120 to -5 dBm.

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. Alternatively, the traffic channel, together with other parameters, can be changed in the **Hard Handoff** menu. The valid input range depends on the selected frequency band.

Band — The frequency band which is used to transmit the CCH or TCH. The channel number together with the frequency band identify the carrier frequency.

RC — Radio configuration. The radio configuration determines the supported traffic channel data rates and modulation types on the forward and reverse link. In the measurement menus, the radio configuration can be viewed only. To change the radio configuration and service options in idle mode, go to the **Parameters > Traffic Channel** menu. You can change these parameters in the **Hard Handoff** menu while on a traffic channel. A complete list of radio configurations and service options is provided in section ["Radio configurations, service options" on page 97](#).

SO — Service Option. The service option determines the mode in which the traffic channel will operate, voice or loopback. In the measurement menus, the service option can be viewed only. To change the radio configuration and service options in idle mode, go to the **Parameters > Traffic Channel** menu. If you want to change these parameters while a traffic channel is assigned, enter the **Hard Handoff** menu.

SID — The System Identity (SID) is a number which identifies the subscriber's home system. Using a home SID allows operation with phones that are programmed for home-only operation. In the measurement menus, the system identity can be viewed only. To change the SID, change to **Parameters > Network Setup**.

NOTE

Many other test parameters as well as the limits for the various tests can be modified using the [Forward link](#) menus.

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. This is only displayed for measurements, which do not have a bar display associated with the measurement (Open Loop and Gated Power). |
| FAIL | This mark indicates that test limits have been violated by the measurement results. This is only displayed for measurements, which do not have a bar display associated with the measurement (Open Loop and Gated Power). |
| ---- (Four hyphens) | Will be displayed when the measurement has not been started yet or when there are no limits available for the current test or there are multiple measurements being performed on the current screen. |
| POWER, MIN, MAX | When performing the "Min/Max" power test in CDMA2000 mode, this will indicate which test is currently being performed. |
| STANDBY, ACCESS | When performing the "Access Probe" in CDMA2000 mode, this will indicate which test is currently being performed. |




Status

Signaling status from the 4400's point of view.

| | |
|----------------------------|---|
| 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. |
| ----- (six hyphens) | This indicates that the call setup has not been started yet or that there has been no communication between the 4400 and the mobile so far. |

The large display field below provides more details and a short signaling history. On some of the menus, you can even scroll back within the history using the **PgUP**, **PgDn**, **UP** and **DN** keys.





Display markers

| Marker | us | dB |
|---|------|----|
|  | 20 | |
|  | 1230 | |
|  | 1210 | |

The 4400 offers two markers for easy reading of the graphical displays. On the input fields in the first column you set the markers to the required horizontal positions. The corresponding result fields in the next column shows the measurement values obtained for the respective positions.

The third row shows the difference between the markers in both time and measured value.

Statistical test evaluation

| | Curr. | Min. | Avg. | Max. | Count | 24 |
|-------|--------|--------|--------|--------|---|---|
| Upper | 25.49 | 24.52 | 26.18 | 28.23 |  |  |
| Lower | -50.83 | -51.16 | -49.90 | -46.71 |  |  |

The following information can be found in the results table:

- Count** On some of the test menus, the 4400 provides you with statistical evaluations of the measured values. Those statistical 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.
- Curr.** Current measurement value. The last measurement obtained without any statistical evaluation.
- Min.** The absolute minimum of all measurements obtained.
- Avg.** The average value calculated using a block of measurement values. See below for more information about averaging.
- Max.** The absolute maximum of all measurements obtained.

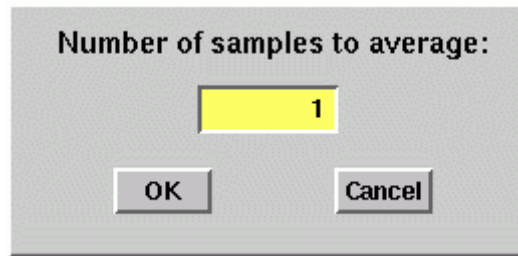
NOTE

The bar display is described in section ["Display markers" on page 41](#).

NOTE

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

The average value on the results screens is determined by the last N measurements. The number N can be selected with the "Number of samples to average" input box which is available on some (not all) of the measurement menus; to access this field, push **ENTER** on the Avg. input field or click on it with the mouse. The following input box pops up:



You can enter any number from 1 to 100. Using the **RIGHT** cursor key, move the cursor to the OK button and push **ENTER**; alternatively, use the mouse to click on the OK button.

Softkeys in idle mode

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 a set of parameter menus on-screen. With the help of these menus, 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** when pressed.

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 43.

Register

This softkey forces the mobile phone to register with the network. The result is that the identity parameters (ESN, MIN) are updated and results from previous measurements are deleted. You can use this softkey to update the phone parameters in the 4400 when you start testing a new phone; this includes information about the phone's capabilities as displayed in the MS Info menu.

Note: The mobile is forced by simulating a cell with information different from that previously used. According to CDMA standards, the mobile has to register with the network when it changes cells.

Reset

This softkey reinitializes the statistical evaluation of the measurements, i.e. results from previous measurements are deleted.

MS Call When the mobile phone is registered with the base station (the 4400) and idle, you may always dial a number and push the 'send' button on the mobile to start a mobile-originated call.

While the call is being established, the [Status](#) section of the menu displays the corresponding status information. 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 43 for details.

BS Call A push on this softkey initiates a BS-originated call setup. The base station will always call the number of the phone that has registered last with the 4400.

In case the mobile does not answer the BS call, it may not be registered to the network simulated by the 4400. Usually, the phone registers automatically. Possible scenarios which might inhibit the phone from registering with the 4400 include:

- The phone found a real network to register with.
This can be avoided by placing the phone in a shielded environment, e.g. in the Willtek RF Shield.
- The phone does not support the frequency band or the channels that the 4400 is using.
Verify the Band and CCH setup and ensure that these are supported by the phone under test.
- The mobile phone does not support roaming and the 4400 simulates a network other than the mobile's home network.
Change the 4400's SID (System IDentity) in the **Parameters > Network Setup** menu to the mobile phone's home SID.

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 [Forward link](#) menu on-screen. With the help of this menu, you may view test parameters, limits, and the gated power/time template.

Release Call

A push on this softkey terminates the call.

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.

Zoom

The **Zoom** softkey toggles through different scale views.

Note: This softkey is only available in the Gated Power test menu.

**Toggle
Test Type**

The Toggle Test Type appears in a few test menus to change the measurement type. See in the description for the respective test menu for more details.

Execute

The **Execute** softkey initiates an action. This softkey is only available in the Hard Handoff menu.

Testing the phone on CDMA channels

Your 4400 comes with many clever thought-through features that keep CDMA 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.

The structure of this chapter reflects the different test menus available with the marker tabs:

- [Testing voice or data?](#)
- [Basic test menu](#)
- [Rho/Frequency test menu](#)
- [Overview of power tests](#)
- [Min./max. power test](#)
- [Open loop power control test menu](#)
- [Gated power test menu](#)
- [Closed loop power control menu](#)
- [Access probe menu](#)
- [FER test menu](#)
- [Code Domain test menu](#)
- [MS Info test menu](#)
- [Hard Handoff menu](#)

Testing voice or data?

Before you start measurements, be clear about the parameters and channel types you want to test.

If you want to test RF parameters and voice quality as well as the basic operation of the CDMA phone, we recommend to use a voice channel for testing. Voice channels allow transmitter and receiver measurements as well as supporting a loopback function for qualitative voice tests.

If you want to test the enhanced cdmaOne and CDMA2000 features such as transmission and reception at higher data rates and with multiple code channels, data channels are recommended for your testing.

The type of channel can be set up as follows:

Table 2 Voice channel vs. data channel

| Channel type | Service options to be used |
|---|----------------------------|
| Voice (signal from mobile phone looped back by the 4400) | 1, 3, 17, 32768 |
| Voice loopback (signal generated in the tester looped back by the mobile phone) | 2, 9, 55 |
| Data | 32 |

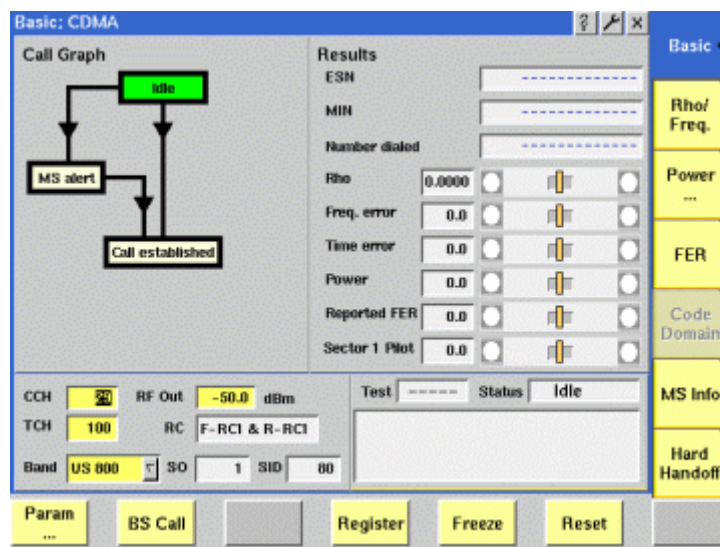
NOTE

Service option 55 is not supported by all mobile manufacturers.

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, this basic test menu is an ideal test-time saver.



The basic test menu provides you with

- a graphical overview of the ongoing signaling
- the most important test results of the mobile's transmitter and receiver in both graphical and numerical formats
- the mobile's ESN and MIN
- the display of the number dialed on the mobile in case of a mobile-originated call setup.

NOTE

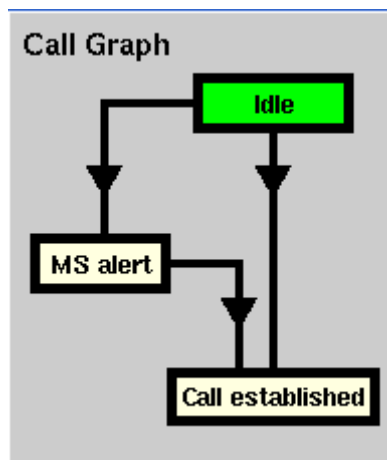
A detailed description of the Test and Status area of this menu can be found in section "Status" on page 40.

NOTE

The meaning of the softkeys is described in section [Softkeys at call level](#).

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 to push the **BS Call** 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 an MS-originated call or through the MS alert state during a BS-originated call setup.



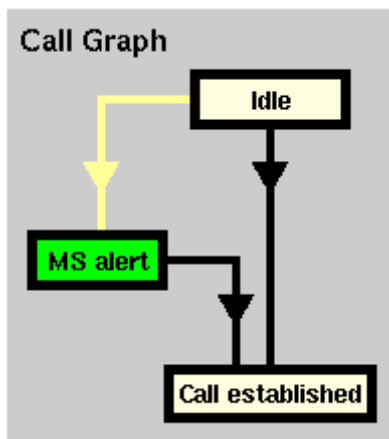
All rectangular boxes with white background indicate states that can be reached using the paths.



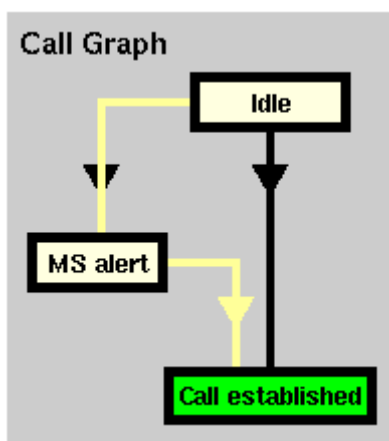
The orange line with the arrow indicates the path that has been used to achieve the current state.

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. The basic CDMA measurements will be performed automatically (for details, see section ["Basic test results display" on page 49](#)).

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 in idle mode" on page 42](#)).

MS-originated call

To initialize a MS-originated call, 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.

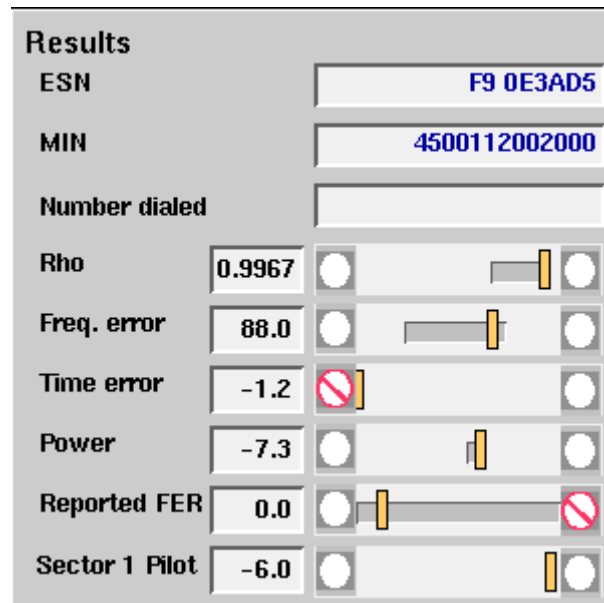
As soon as the call has been put through, you are working on call level. The basic CDMA measurements will be performed automatically (see section "Basic test results display" on page 49 for details).

Basic test results display The basic test results display gives a short overview of the mobile's most important characteristics. Two different versions of this display exist.

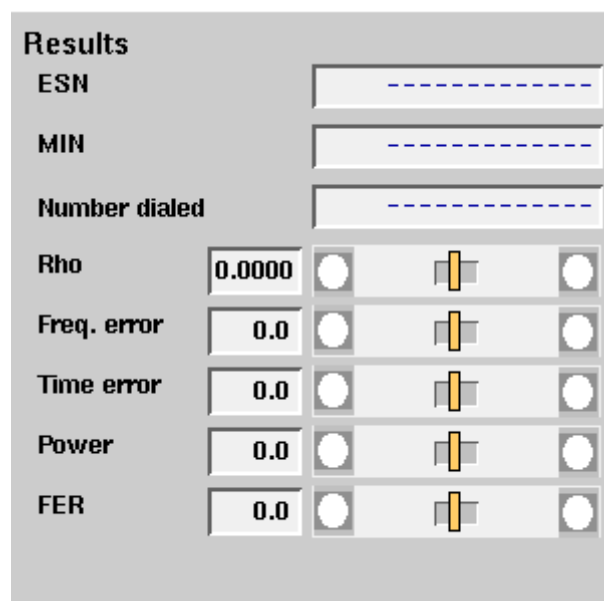
NOTE

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

Basic test results display in voice mode:



Basic test results display in loopback mode:



How to read the basic test results display

| | |
|----------------------|---|
| ESN | <p>The Electronic Serial Number (ESN) uniquely identifies the mobile phone. It consists of two fields – the manufacturer code and the serial number. The two values are displayed in this field as hexadecimal numbers.</p> <p>Both values are displayed as decimal numbers in the MS Info menu.</p> <p>The ESN display appears when the phone registers with the 4400 or when a call is being set up.</p> |
| MIN | <p>The Mobile Identification Number (MIN) identifies the subscriber. It is a ten-digit decimal number.</p> <p>The MIN display appears when the phone registers with the 4400 or when a call is being set up.</p> |
| Number dialed | <p>Following an MS-originated call, the 4400 displays the number that was initially dialed on the mobile.</p> <p>This display field will remain empty unless the call has been originated by the mobile.</p> |
| Rho | <p>Rho is the waveform quality factor, a measure of the modulation accuracy. The waveform quality factor is indicated and frequently updated during an ongoing call. A value of 1 indicates perfect waveform quality.</p> <p>The limits for this test are set on the Traffic channel menu. IS-98 specifies that Rho must be greater than 0.944.</p> |
| Freq. error | <p>Here, the 4400 displays the results of the statistical evaluation of the frequency error measurements. The result is given in Hertz.</p> <p>IS-98 specifies that the frequency error must be within ± 300 Hz.</p> |
| Time error | <p>The time error indicates the transmit time error (Tau) with respect to the forward channel timing and is displayed in microseconds.</p> <p>IS-98 specifies that the time error must be within ± 1 μs.</p> |
| Power | <p>Power indicates the power measurement and is displayed as the mean output power of the mobile station's transmitter. Transmitter output power control in the CDMA system is complex and critical. Refer to "Radio configurations, service options" on page 97 for testing the various components of CDMA power control.</p> <p>The limits for this test are set on the measurement limits menu. This menu is a submenu of the Forward link menu.</p> |
| Reported FER | <p>Reported FER (frame error rate) is available with voice channels without loopback only and indicates the number of bad frames (frame errors) the mobile has received relative to the number of frames received. The mobile reports these figures to the 4400 when the service option is set to 1, 3, or 32768.</p> |

- Sector 1 Pilot** This field indicates the pilot strength reported by the mobile. The value is relative to the complete carrier signal and expressed in dB. The mobile reports these figures to the 4400 when the service option is set to 1, 3, or 32768.
- FER** The FER – or frame error rate for short – is a measure for the receiver quality or sensitivity. The FER is measured at a defined RF power level (4400 transmit, mobile receive power). The FER is given as a percentage of all frames transmitted and received.

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](#) menu in the basic parameters section.

Other screen elements

Read section ["Softkeys in idle mode" on page 42](#) to learn how to view and modify channels, power level and other call-related parameters.

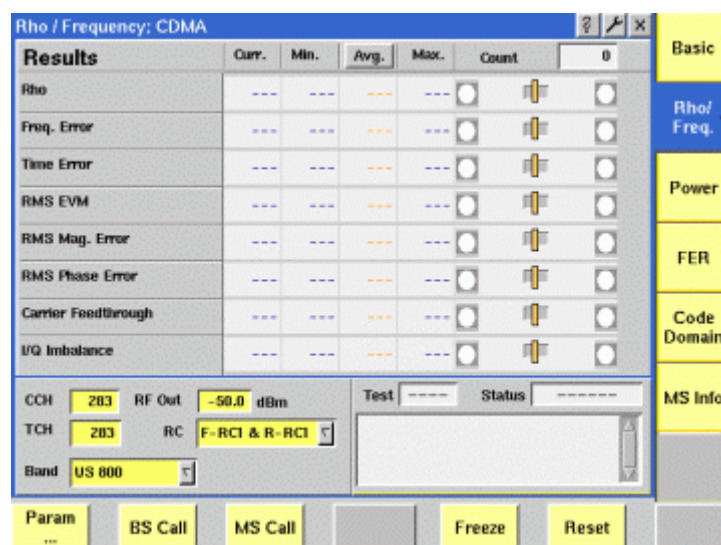
Section ["Status" on page 40](#) explains the elements of the status area.

The ["Statistical test evaluation" on page 41](#) tells you how to read the Min/Max/Avg tables.

The functions of the softkeys depend on whether or not a call has been established. See sections ["Softkeys in idle mode" on page 42](#) and ["Softkeys at call level" on page 43](#) for the available softkey functions.

Rho/Frequency test menu

This test menu gives you all relevant parameters required to judge the modulation quality of the mobile.



NOTE

The meaning of the softkeys is described in section "Softkeys at call level" on page 43.

NOTE

Need to know more about the Test and Status area of this menu? All details can be found in section "Status" on page 40.

The Rho/frequency results display shows the numeric results and the statistical evaluation of the test.

| Results | Curr. | Min. | Avg. | Max. | Count | 0 |
|---------------------|-------|------|------|------|-------|---|
| Rho | --- | --- | --- | --- | | |
| Freq. Error | --- | --- | --- | --- | | |
| Time Error | --- | --- | --- | --- | | |
| RMS EVM | --- | --- | --- | --- | | |
| RMS Mag. Error | --- | --- | --- | --- | | |
| RMS Phase Error | --- | --- | --- | --- | | |
| Carrier Feedthrough | --- | --- | --- | --- | | |
| I/Q Imbalance | --- | --- | --- | --- | | |

How to read the rho/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.

- Rho** Rho is the waveform quality factor, a measure of the modulation accuracy. The waveform quality factor is indicated and frequently updated during an ongoing call. A value of 1 indicates perfect waveform quality. The limits for this test are set on the [Traffic channel](#) menu. IS-98 specifies that Rho must be greater than 0.944.
- Freq. Error** Here, the 4400 displays the results of the statistical evaluation of the frequency error measurements. The result is given in Hertz. IS-98 specifies that the frequency error must be within ± 300 Hz.
- Time Error** The time error indicates the transmit time error (Tau) with respect to the forward channel timing and is displayed in milliseconds. IS-98 specifies that the time error must be within ± 1 μ s.

| | |
|----------------------------|--|
| RMS EVM | <p>The root mean squared error vector magnitude is a measure of the modulation accuracy at the decision points. The error vector magnitude (EVM) is the difference between the vector of the received signal and that of the ideal (expected) signal. The RMS value is an average of all the decision points (symbols) over a whole measurement interval.</p> <p>EVM is given as a percentage.</p> <p>Note: The error vector can be broken down into a magnitude and a phase component; see magnitude and phase errors below.</p> |
| RMS Mag. Error | <p>The root mean squared magnitude error is a measurement of the error in the mobile's transmit signal size (magnitude) at the decision points.</p> <p>The magnitude error is given as a percentage.</p> |
| RMS Phase Error | <p>The root mean squared phase error is a measurement of the phase component of the vector error of the mobile's transmit signal at the decision points.</p> <p>The phase error is given as a percentage.</p> |
| Carrier Feedthrough | <p>The carrier feedthrough 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.</p> |
| I/Q Imbalance | <p>The I/Q imbalance measurement is an indication of the accuracy of the I/Q modulator's modulating signal amplitude balance. It is the ratio of the power in the desired sideband carrier produced and the undesired sideband carrier produced due to an amplitude difference between the input signals to the I/Q modulator.</p> <p>The I/Q imbalance is measured in dBc.</p> |



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

Overview of power tests

Power control is an important measure for optimum CDMA network performance. Multiple power measurements are defined to check proper behavior of the mobile phone in the real network.

The following power test menus are available:

- ["Min./max. power test" on page 54](#) – shows the minimum and the maximum power level at which the mobile phone is able to transmit.
- ["Open loop power control test menu" on page 56](#) – displays the mobile transmit power/time response to an abrupt change in mobile input level.
- ["Gated power test menu" on page 57](#) – shows the power versus time profile when the mobile phone transmits gated (burst) signals.

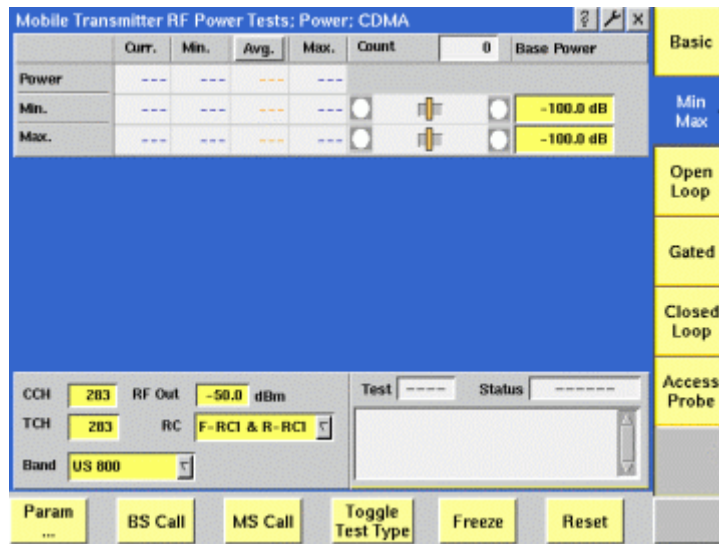
- "Closed loop power control menu" on page 60 – tests the mobile phone's ability to perform closed loop power control and measures the mobile's closed loop dynamic range.
- "Access probe menu" on page 62 – measures the power of the first access burst and the residual power in standby (idle) mode.
- "Code Domain test menu" on page 68 – displays the power of the code channels and the in-phase (I) and quadrature (Q) signals.

The code domain power measurements can be accessed from the CDMA basic menu by pushing the [Code Domain] marker tab.

For the other power measurements, push the [Power...] marker tab to access these power tests. Push the marker tab for the wanted power measurement or [Basic...] to return to the CDMA basic menu.

Min./max. power test

From the CDMA basic menu, push the [Power...] marker tab to enter the Min Max (Mobile Transmitter RF Power Tests) menu.



Function

This menu shows three different types of measurement. Each measurement type represents the mobile transmit power. The types of measurement are referred to as the Power measurement (affected by the RF Out level and the currently selected power control mechanism), the Min. and the Max. measurement (taken at the respective base power level, with power control set to Up or Down, respectively). There is only one type of measurement being performed at any time.

The Min. measurement represents the minimum power level supported by the mobile phone. Setting the phone to its lowest power level is achieved by a combination of open loop power control (where the 4400's output power is set to a high level) and closed loop power control (the mobile is repetitively told to decrease its power level).

The Max. measurement represents the maximum power level supported by the mobile phone. The high power level of the phone is the result of the 4400's output power being set to a low level and the closed loop power control commanding the phone to increase its output power.

The Power measurement is the current power level, which depends on the power control mechanism selected in the Common Parameters section in the **Parameters > Traffic Channel** menu and on the power level that the mobile phone receives from the 4400. The higher the RF Out (base transmit level) value, the lower the mobile's transmit power will be.

NOTE

In the Min. and Max. measurements, closed loop power control is set to Up or Down, respectively. This is done automatically by the 4400 and overrides the manual selection of the power control mechanism. Once you select the Power test type or leave this menu, the 4400 returns to the manual power control selection.

Selecting the measurement type

Use the **Toggle Test Type** softkey to change between Power, Min. and Max. measurements. The 'Test' output field indicates the measurement type currently selected.

When Test is set to Power, the 4400 will apply the output power level selected in the RF Out input field.

How to read the mobile transmitter RF power 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 relative to the measurement limits.

| | |
|--------------|--|
| Power | This is the overall power level (in dBm) transmitted by the mobile phone and measured by the 4400. The level depends on the RF Out setting. Results are updated only when the test has been selected with the Toggle Test Type softkey. |
| Min. | The minimum power level (in dBm) is measured when the Min. test has been selected. |
| Max. | The maximum power level (in dBm) is measured when the Max. test has been selected. |

| | | |
|--------------|----------|---|
| Count | 0 | Shows the total number of measurements used to calculate the statistical results. |
|--------------|----------|---|

Selecting measurement-specific parameters

Min. Base Power — This is the power level (in dBm) received by the mobile phone. If no coupling loss compensation is active, this is the RF Out level of the 4400. The default value is -25 dBm.

Max. Base Power — This is the power level (in dBm) applied to the mobile phone receiver. If no coupling loss compensation is active, this is the RF Out level of the 4400. The default value is -104 dBm.

Min. limits — To change the upper limit for the minimum power, go to the **Parameters > Limits > Limits Cont.** menu and select the Minimum Power parameter.

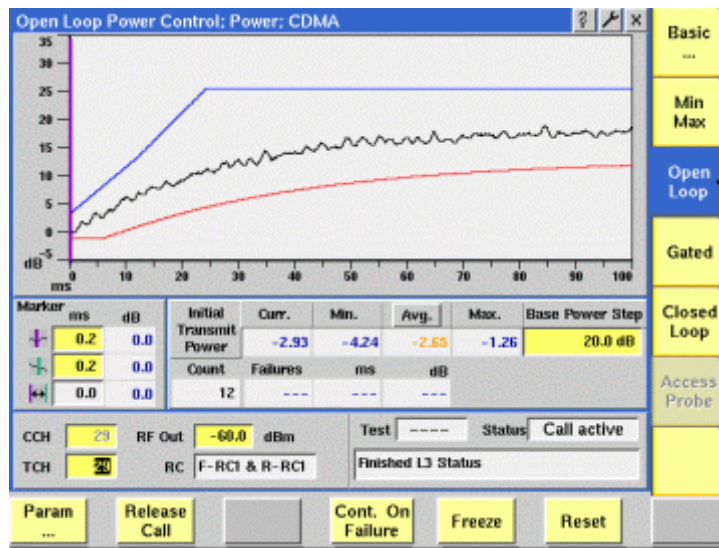
There is no lower limit for Min. power.

Max. limits — The lower and upper limit for Max. power depends on the mobile's power class. To change the lower limit, go to the **Parameters > Limits > Limits Cont.** menu and select the corresponding power class of the ERP matrix. The 4400 uses the limit for the power class that the mobile phone reports (see MS Info menu).

Power Control — The mechanism for closed loop power control is determined in the Common Parameters section in the **Parameters > Traffic Channel** menu.

Open loop power control test menu

To check the open loop power control mechanism of the mobile phone, push [Power] > **Open Loop** from the CDMA basic menu.



NOTE

This measurement requires either of the loopback service options (2, 9, or 55), otherwise no results will be shown. The service option can be set up in the **Parameters > Traffic Channel** menu.

Function

The open loop power control test menu visualizes the mobile phone's ability to cycle power up or down after the received power level is stepped down or up, respectively. It is a measurement of the time response of the open loop power control; the phone is required to ramp its transmit power up or down within a certain time, which is measured and displayed here. The measurement is repeated i.e. the power step is applied in both directions; only one direction is shown, however.

The size and direction of the power step is user-definable; the step is exercised periodically. The traffic channel is forced to full rate for the time the Open Loop menu is selected.

NOTE

The traffic channel data rate is forced to Full for this measurement. When you leave this menu, the 4400 will set the data rate back to the user-defined setting.

How to read the open loop power control test results display

The graphical display shows the RF power versus time, together with the upper (blue line) and lower (red line) limits. The limits are configurable in **Param > Limits > Open Loop**.

Use the **Display markers** for exact measurement result readings at defined points in time.

The statistical evaluation of the measurement results is displayed as current, minimum, average and maximum values.

**Initial
Transmit
Power**

The power level that the mobile phone transmits before the power level step. The measurement is repetitive and before each step, the mobile's transmit power returns to about the same value as initially. This measurement can be used to verify that each step starts at about the same power level.

NOTE

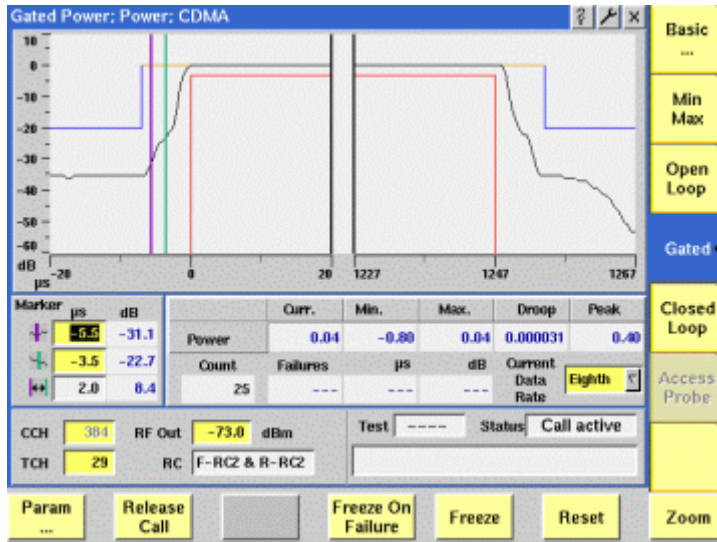
The power vs. time result may be outside the defined limits when the mobile phone already transmits at or near its minimum and maximum supported power level. This is not a malfunction of the mobile phone.

Selecting measurement-specific parameters

Base Power Step — Determines the difference in 4400 output power that should initiate a variation of the mobile output power. Valid inputs are in the range from -30 to +30 dB.

Gated power test menu

From the CDMA basic menu, push [Power] > **Gated** to get to the gated power menu.



Function

When the gated power test menu is selected, the 4400 forces the mobile phone to transmit at a subrate so that the phone only transmits every second, fourth or eighth power control group or burst (i.e. it gates its transmit signal). This allows the user to examine the gated signal (power vs. time) against the specified template. The graphical display shows the gated power versus time, averaged over a number of transmissions (power control groups or bursts). The number of measurements over which the 4400 averages the results can be defined on the **Parameters > Limits > Gated PTT** menu.

You can select the gating rate using the **Current Data Rate** selection field. Once another test menu is selected, the 4400 returns to the data rate selected in the **Parameters > Traffic Channel** menu.

Note that the transmission time and the template depends on the selected radio configuration.

The template for gating can be defined for the different radio configurations in the **Parameters > Limits > Gated PTT** menu.

How to read the gated power test results display

The graphical display shows the mobile's transmit power versus time for one gated-on period (burst). The horizontal scale is microseconds and the vertical scale is dB relative to the average power for the power control group. The upper and lower limits are indicated by straight lines in red and blue. The limits can be changed by pushing **Param > Limits > Gated PTT**.

Curr./Min./ Max. Power

These output fields indicate the RF power level (in dBm) over a gated-on period (burst). The statistical evaluation of the measurement results is displayed as current, minimum, and maximum values.

Droop Power

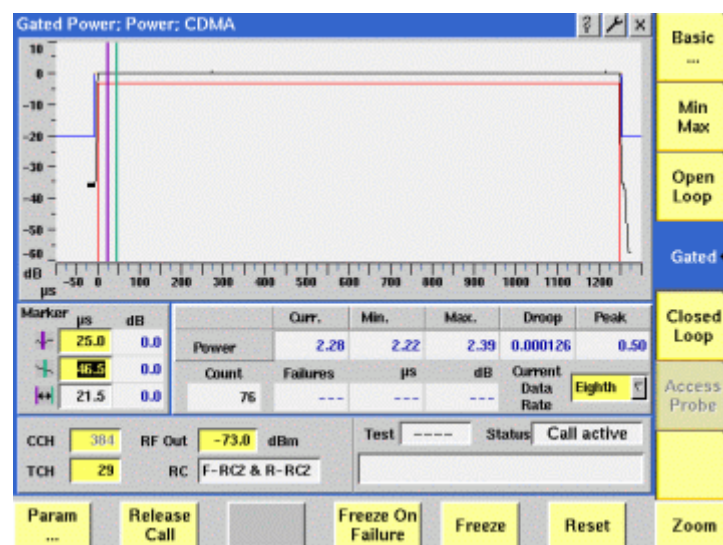
Droop is the average change in amplitude per symbol. The unit is dB/sym.

- Peak Power** This result field indicates the peak power over the gated-on period last measured.
- Count** This field represents a counter of the measurements already taken and represented in the statistical evaluation of the power results. A push on the Reset softkey clears the memory of both the statistical evaluation and the counter.
- Failures** If the present measurement result of power versus time fits within the power/time template, this field indicates a PASS, otherwise it indicates FAIL.
- μs** This field indicates the position of the first failure against the power/time template.
The μs field indicates the first occurrence of a failure of the measurement relative to the template. If you press the "Freeze on Failure" softkey, you will be able to analyze this. After a failure occurs, you may go to one of the markers and enter the same value as displayed in the μs field; this will place the marker at the first point in the data which exceeds the limit template.
- dB** If the power versus time fails the template, the maximum deviation of the power level from the template is shown here.

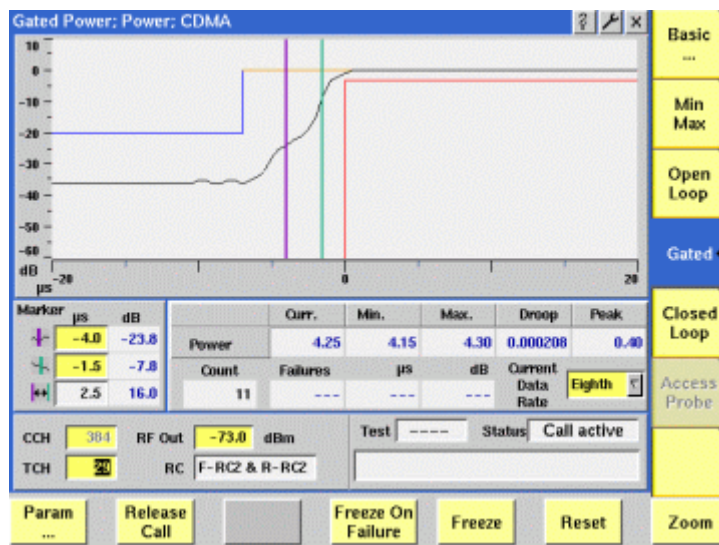
Viewing certain details of the gated power

The initial graphical representation of the gated power shows the power versus time at both edges of the burst. You can view other parts of the burst by pushing the **Zoom** softkey.

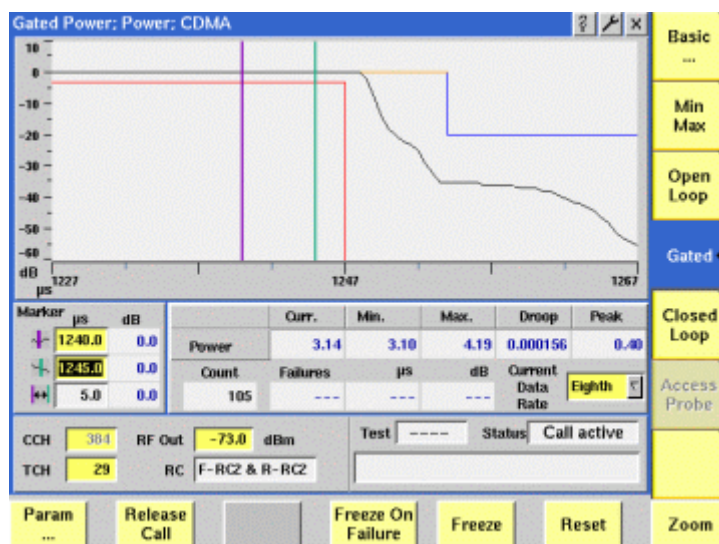
Upon the first push on the **Zoom** softkey, the power over the whole power control group is displayed (see example below).



Upon the second push on the **Zoom** softkey, the power at the rising edge is shown.



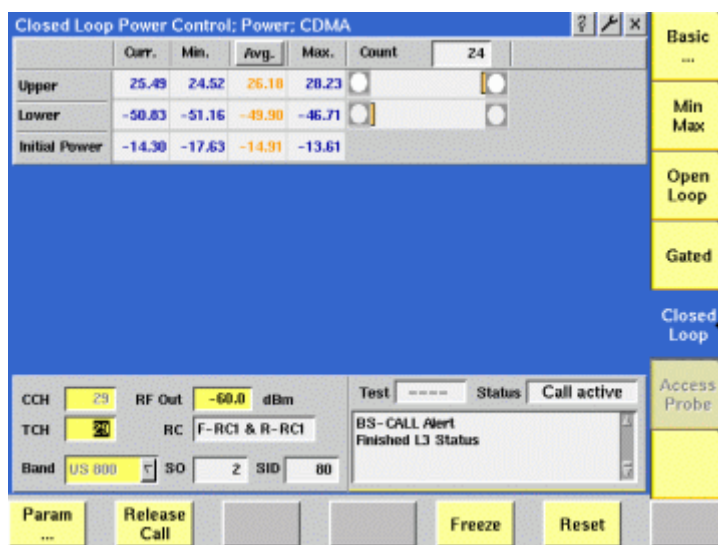
After pushing the **Zoom** softkey a third time, the falling edge is shown.



The fourth push returns to the basic gated power display.

Closed loop power control menu

From the CDMA basic menu, push [Power] > **Closed Loop** to access the closed loop power control menu.



NOTE

This measurement requires either of the loopback service options (2, 9, or 55), otherwise no results will be shown. The service option can be set up in the **Parameters > Traffic Channel** menu.

Function

This measurement menu can be used to verify the mobile phone's closed loop power control ability and to measure its maximum and minimum power level, respectively.

The measurement procedure starts when entering the closed loop power control menu:

The power is reset to -15 dBm. This is the initial value at which a measurement of the power level is taken in compliance with IS-98D test 4.4.4 step 9. The 4400 then directs the CDMA phone to increase its power level by means of sending 112 power control bits set to "0", or up. Once the mobile phone should be transmitting at its maximum power level, the 4400 then sends 100 power control bits set to "1" or down. After the power control bits have been transmitted, the 4400 measures the lower power. The 4400 then sends 100 power control bits set to "0" or up. The 4400 then measures the upper power after the power control bits have been transmitted. Finally, the initial power (in dBm), the upper and lower power levels (in dB, relative to the initial power) are displayed.

The 4400 repeats this process, each time updating the screen with three new results.

The closed-loop power control mechanism in this menu depends on the test being performed rather than on the entry in the corresponding field in the **Parameters > Traffic Channel** menu. The 4400 switches back to the user-defined value when selecting another menu.

How to read the closed loop power 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 relative to the measurement limits.

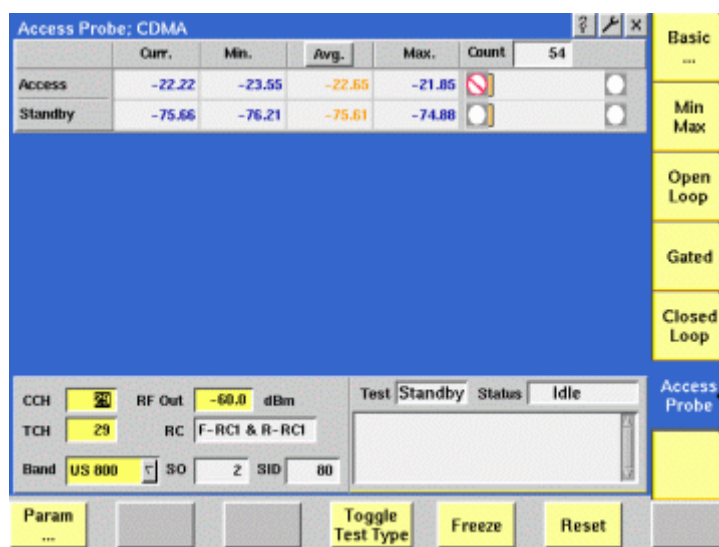
The Upper result field represents the relative power measurement result after the phone has been ordered to the highest power level.

The Lower result field indicates the relative power level result after the phone has been ordered to the lowest power level.

The Initial Power result field shows the power level at each start of a new cycle, before the power level is increased and decreased. No bar display is included since there are no limits for this measurement.

Access probe menu

This menu is accessible only when the mobile is in idle mode, i.e. not assigned to a traffic channel. From the CDMA basic menu, push [Power] > **Access Probe** to get to the access probe menu.



Function

This menu can display the power level during the initial access probe (first burst) while the mobile phone accesses the network, e.g. to set up a call. It can also display the power level emitted while in standby (idle mode). The tests can only be performed when the mobile phone is in idle (standby) mode.

In order to measure access probes from the mobile phone, the 4400 sends paging messages. The mobile phone sends a sequence of access probes, increasing the power level with each access probe of a sequence. The 4400 does not respond to the access probes and hence the mobile phone sends more and more access probes sequences, allowing the 4400 to take measurements repetitively. Note that only the first access probe within an access sequence is measured; the first (or initial) access probe is transmitted at the lowest power level.

How to read the access probe 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 relative to the measurement limits.

| | |
|----------------|--|
| Access | This is the measured power (in dBm) of the first burst of an access probe. Measurements are updated frequently when the Access test has been selected using the Toggle Test Type softkey. |
| Standby | The standby power (in dBm) is the in-band power level measured while the mobile phone is inactive and observing the base station signals. Measurements are updated frequently when the Standby test has been selected using the Toggle Test Type softkey. |

Selecting the measurement type

Use the **Toggle Test Type** softkey to change the measurement between Access power and Standby power. Only one of the two measurement types can be performed at any time. The 'Test' output field indicates the measurement type currently selected.

Selecting measurement-specific parameters

The following parameters affect the access probes, e.g. the initial power level or the power steps in which the power is increased with each new access probe within an access sequence:

- Nominal Power
- Initial Power
- Power Step
- Probe Steps
- Response Sequences

These parameters can be set up in the Access Channel Parameters section in the **Parameters > Network Setup** menu.

FER test menu

The FER test menu is utilized to measure the quality of the mobile's receiver. The results provided are based on a statistical analysis over a number of frames (bursts). The acronym FER stands for frame error rate.

Two different FER measurements are supported. If the mobile is in voice service, the Reported FER measurement is performed. Otherwise, if the mobile is in loop-back service, the 4400 performs an FER measurement. The table below summarizes the FER modes with the applicable service options.

NOTE

When the call is mobile-originated, the mobile usually requests a voice service option, resulting in Reported FER measurements. For FER measurements, it is advisable to select one of the applicable loopback service options and perform a base station-originated call.

Table 3 FER modes

| Reported FER (voice service) | FER (loopback service) |
|---|---|
| SO 1 (basic variable rate voice, 9.6 kbps) | SO 2 (mobile station loopback, 9.6 kbps) |
| SO 3 (enhanced variable rate voice, 9.6 kbps) | SO 9 (mobile station loopback, 14.4 kbps) |
| SO 17 (high rate voice, 14.4 kbps) | SO 55 (data loopback) |
| SO 32768 (basic variable rate voice (14.4 kbps) | |

The FER tests can be performed under different conditions; the measurement result heavily depends on these conditions:

- Absolute power of the CDMA signal
- Power of the channel(s) under test relative to the overall simulated signal
- Presence and strength of a CDMA signal from a second simulated antenna sector
- Presence and strength of additive white Gaussian noise (AWGN)

Reported FER (voice service)



Function

This menu is used for voice service options; the mobile frequently sends messages informing the 4400 of the number of bad frames received. If the mobile phone cannot decode the received speech frame then it marks it as bad; every

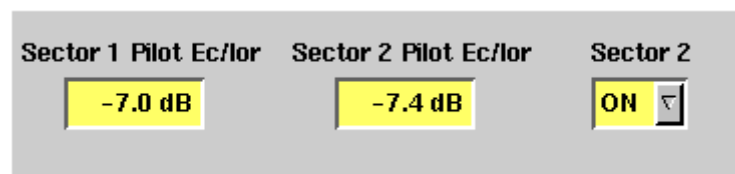
160 frames it reports the number of bad frames to the 4400. The 4400 compares the number of speech frames transmitted to the mobile phone with the number of frames the phone received in error, and calculates the frame error rate.

How to read the Reported FER test results display

The **Reported frame error rate (FER)** is the ratio of the voice frames undecodable by the mobile relative to all voice frames transmitted to it. The Reported FER calculation is based on data that the mobile reports to the tester.

| | |
|-----------------------|--|
| FER | This is the final result after comparing the transmitted frames with the received ones. The frame error rate is expressed as a percentage. The bar graph displays a fail when the user-defined FER limit is exceeded. The limit can be defined in the RX Sensitivity section of the Parameters > Limits menu. |
| Frames TX | This field gives the number of frames transmitted to the phone. Interim results are shown while the measurement is going on. The total number of frames transmitted to the phone to give the result for one measurement is 160. |
| Frame Errors | This is the number of frames indicated by the mobile phone to be erroneous. |
| Sector 1 Pilot | The mobile phone reports its measurement (in dB) of the pilot channel signal strength. The result can be compared to the user-defined value (Sector 1 Pilot Ec/Ior). Sector 1 Pilot is the signal strength of the sector 1 pilot relative to the overall power of the signal transmitted for the simulated base station antenna sector 1. |
| Sector 2 Pilot | The mobile phone reports its measurement (in dB) of the pilot channel signal strength. The result can be compared to the user-defined value (Sector 2 Pilot Ec/Ior) if a second sector is being simulated by the 4400. Sector 2 Pilot is the signal strength of the sector 2 pilot relative to the overall power of the signal transmitted for the simulated base station antenna sector 2. |

Setting up Reported FER measurement parameters



Sector 1 Pilot Ec/Ior — The Sector 1 Pilot Ec/Ior input field defines the energy of the pilot channel accumulated over one PN chip period (Ec) relative to the total transmit power spectral density for this sector. The default value is -7.0 dB.

Sector 2 Pilot Ec/Ior — The Sector 2 Pilot Ec/Ior input field defines the energy of the pilot channel accumulated over one PN chip period (Ec) relative to the total transmit power spectral density for this sector. The default value is -7.0 dB. This input field is available only if sector 2 is in use.

Sector 2 — This selection field can be used to turn the simulated sector 2 on or off. The default is off. If sector 2 is not in use, the pilot Ec/Ior for this sector cannot be entered.

FER (loopback service)



Function

In loopback service, the 4400 transmits speech or data frames to the mobile phone; the phone transmits the received data back to the 4400. The 4400 presents the FER as the ratio of erroneous frames to the complete number of frames looped back.

How to read the FER results display

- FER** The FER (frame error rate) indicates the number of bad frames (frame errors) the mobile has received relative to the overall number of frames received.
- Frames TX** This parameter gives the number of frames that have been transmitted to the mobile. Note that this number may deviate from the user-defined number of frames defined for a measurement since the test may stop before the number of frames has been reached. See explanations for the Status results field.
- Frame Errors** This gives the number of frames that the mobile received and returned in error.

Status

This output field indicates one out of four different results:

- **Pass** if the maximum number of frames has been transmitted and the calculated frame error rate does not exceed the user-defined limit.
- **Pass Conf.** (pass with confidence) if the test has been stopped before the maximum number of frames has been reached; the frame error rate is below the user-defined limit.
- **Fail** if the maximum number of frames has been transmitted and the calculated frame error rate is above the user-defined limit.
- **Fail Conf.** (fail with confidence) if the test has been stopped before the maximum number of frames has been reached and if the frame error rate exceeds the user-defined limit.

Pass or fail with confidence means that the FER test has been stopped before the specified number of samples has been reached. This is done because the measurements so far exceed the user-defined probability (or confidence level) for a pass or fail should the test be carried out to the end. This shortcut helps save test time. Whether or not the measurement will be stopped beforehand, depends on the statistical distribution of results.

Setting up FER measurement parameters

| | | | | |
|------------------|--------------------|-------------------|---------------------------|--------------|
| FER | Conf. Level | FCH Ec/Ior | Ec/Io | EB/Nt |
| 0.5 % | 95.0 % | -7.4 dB | -12.2 dB | 13.7 dB |
| Data Rate | Frames | AWGN | AWGN / CDMA Signal | |
| FULL | 1000 # | ON | 0.0 dB | |

FER — This is the limit value for the FER measurement. If the measurement exceeds this value, the test is failed, otherwise the mobile passes the test. The FER limit value is entered as a percentage; valid inputs range from 0 to 5 in multiples of 0.1.

IS-98 specifies that the mobile's FER shall not exceed 0.5% with 95% confidence level.

Conf. Level — The confidence level is a statistical measure of the completeness of information obtained about a statistical process. A confidence level of less than 100% may result in the test being concluded before the specified number of frames has been reached, so that the test time can be shortened. See "[System parameters](#)" on page 73 for more information.

The valid input range for the confidence level is from 95% to 100% in steps of 0.1%.

FCH E_c/I_{or} — This is the ratio (in dB) of the energy accumulated over one PN chip period (E_c) of the forward fundamental channel, to the total transmit power spectral density of the simulated signal of one base station antenna segment.

E_c/I_o — The ratio (in dB) of the combined (antenna segment 1) pilot energy per chip (E_c) to the total received power spectral density (noise and signals) (I_o). This value cannot be entered but is calculated from other parameters such as the entry for FCH E_c/I_{or} (in the Parameters menu), the AWGN (if activated) and the second pilot (if activated).

E_b/N_t — The ratio (in dB) of the combined received energy per bit to the effective noise power at the mobile phone. This value cannot be entered but is calculated from other parameters such as the entry for FCH E_c/I_{or} (in the Parameters menu), the AWGN (if activated) and the second pilot (if activated).

Data Rate — This selection field determines the data rate depending on the selected radio configuration. Select a data rate from the following choice of values: FULL, HALF, QUARTER, EIGHTH. The meaning of these values is as follows:

Table 4 Data rate (in bps) as a function of the radio configuration

| Data rate setting | Rate set 1 (RC1, RC3, RC4; SO 2, SO 55) | Rate set 2 (RC2, RC5; SO 9, SO 55) |
|-------------------|---|------------------------------------|
| FULL | 9600 | 14400 |
| HALF | 4800 | 7200 |
| QUARTER | 2400 | 3600 |
| EIGHTH | 1200 | 1800 |

The data rate with service option 55 depends on both the data rate setting and the radio configuration.

Frames — This input field defines the maximum number of frames required to complete an FER measurement and get a new result. The valid input range is from 1 to 10000.

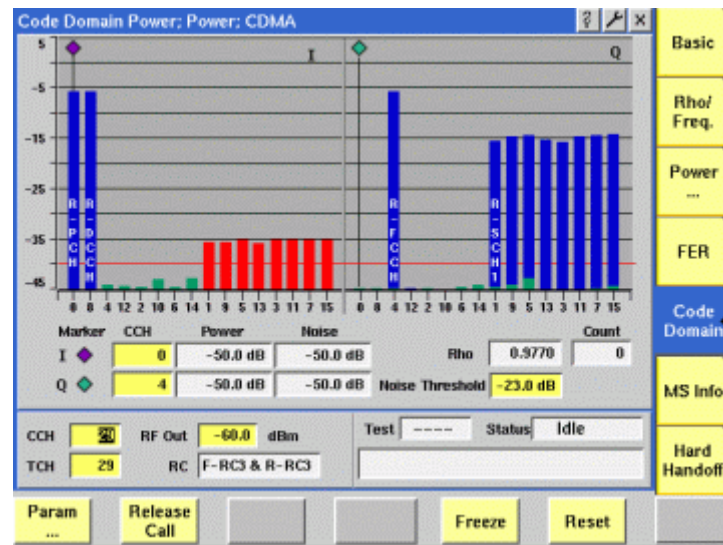
AWGN — This selection field is used to switch Additive White Gaussian Noise (AWGN) on or off.

AWGN/CDMA Signal — This input field determines the signal strength of the additive noise, if switched on, relative to the wanted CDMA signal. The ratio is entered in dB. Note that this value affects the calculated E_c/I_o and E_b/N_t . Valid inputs range from -10 to +5.

Code Domain test menu

When the mobile phone transmits high-speed data using multiple code channels, the Code Domain test menu shows the transmit power for each code channel, relative to absolute power for the complete transmit signal. See ["Test modes" on page 99](#) on how to set up the 4400 so that the CDMA2000 phone transmits multiple code channels.

You can access this menu as long as any of the radio configurations 3, 4, or 5 is selected.



How to read the code domain power results display

The live display shows bars indicating the power. The measurements are separated by both code channels and I/Q proportions. Different colors are used to mark active channels and noise as follows:

- Active channels are shown in blue.
- The code channel noise is colored green. The code channel noise is measured by subtracting the reference signal (i.e., the "ideal" signal) from the received signal, obtaining an error signal, and then despreading this error signal with all of the W_n^{16} codes. This is done for both I and Q components of the error signal.
- If the level of an inactive channel code exceeds the noise threshold, it is colored red.

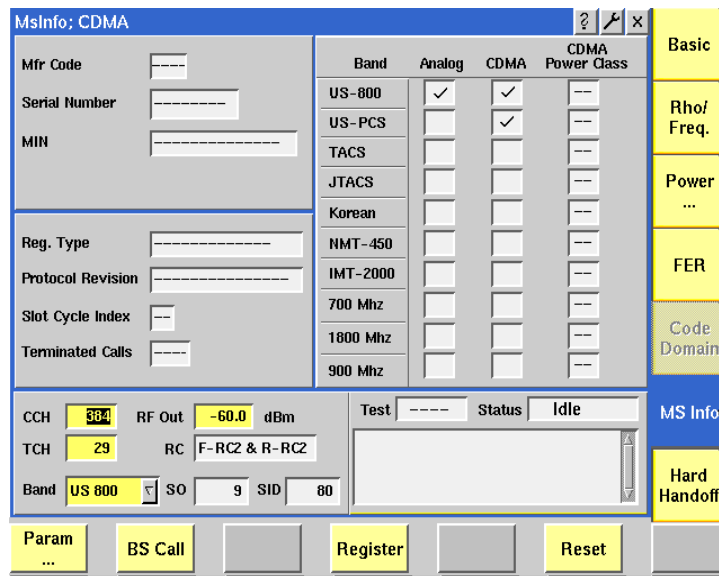
One marker for each phase (I, Q) is available to read out both code channel power and code channel noise.

The waveform quality factor **Rho** indicates the overall modulation quality. Please refer to ["Radio configurations, service options" on page 97](#) for more details on Rho.

Setting up code domain power measurement parameters

The only parameter significant for this menu is Noise Threshold. This value is used as a limit for unacceptably high noise levels. The threshold is indicated as a red horizontal line on the live display; any channel with a noise power exceeding this threshold is indicated by a red bar.

MS Info test menu This menu displays information that the mobile transmits about its capabilities.



Information fields shown in this menu:

- Mfr Code** This is the manufacturer code as reported by the mobile phone.
- Serial Number** This field represents the serial number of the phone. The serial number is assigned by the manufacturer; the combination of manufacturer code and serial number uniquely identifies a mobile phone.
- MIN** The Mobile Identification Number (MIN) is the telephone number of the mobile phone.
- CDMA Power Class** The CMDA Power Class indicates the nominal minimum and maximum power levels that the mobile phone supports in the frequency band currently being used. The power class is indicated as a Roman numeral (I, II, III, IV, or V).
- Reg. Type** This field shows which type of event generated the registration. Possible reasons and outputs are:
 - Timer-based
 - Power up
 - Zoned-based
 - Power down
 - Parameter change
 - Ordered
 - Distance-based
 - User Zone-based

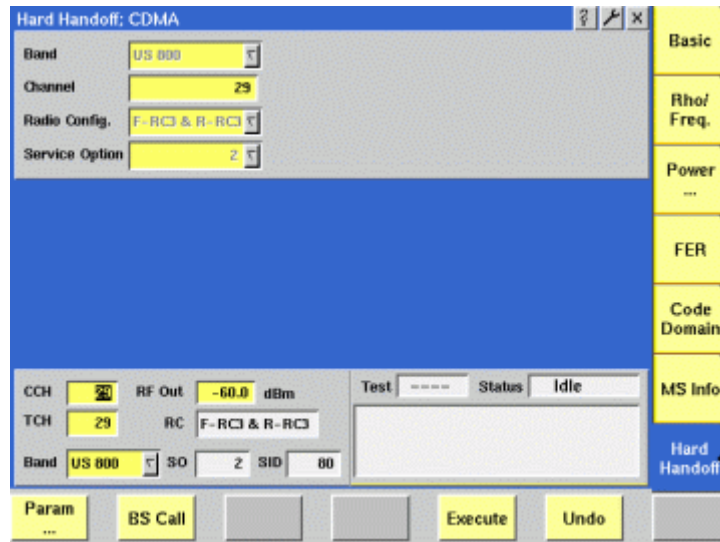
| | |
|--------------------------|---|
| Protocol Revision | <p>The protocol revisions that the mobile may report are shown as follows:</p> <ul style="list-style-type: none">– IS-2000– JSTD-008– TSB-74– IS-95– IS-95A– IS-95B |
| Slot Cycle Index | <p>If the mobile phone is configured for slotted mode operation, this field indicates the preferred slot cycle index (i). If the phone is not configured for slotted mode operation this field is "0".</p> <p>The slot cycle index (i) range is 0 to 7. It specifies how often the mobile in idle state must turn its receiver on and receive the paging channel: it must monitor the paging channel every $2^i * 16$ slots i.e. $2^i * 1.28$ seconds.</p> <p>The mobile phone only monitors the paging channel at specified slots during the slot cycle. This capability offers a trade-off between standby battery life versus access speed. The device can leave the receiver off longer, saving battery life. One extreme are the not so time-critical battery-powered page device controllers that auto answer. In that case a 163.84 second page delay ($2^7 * 1.28$) may not be too long. The device specifies a preferred slot cycle; that is what the 4400 reports in this field. The base station (4400) specifies a max. slot cycle. The lower of the two is what the mobile phone and the network use. The 4400 always sets max. slot cycle to $i=0$.</p> |
| Terminated Calls | <p>Indicates whether the phone is configured to accept calls while operating with the current roaming status. Possible outputs are:</p> <ul style="list-style-type: none">– Yes– No |

The matrix on the right half of the screen provides a summary of the mobile's capabilities. It displays the bands supported, the technology supported within each band and the power class supported for each band.

Hard Handoff menu

The Hard Handoff menu allows you to change the frequency band and traffic channel, or to modify the radio configuration and/or service option. Traffic channel handoffs can be useful in testing the mobile phone on multiple frequencies, e.g. at the lower and upper end of the frequency band supported, without losing time for terminating the ongoing call and setting up a new one on the other frequency. The same considerations apply to changing the radio configuration or the service option when you want to test the phone at different modulation types or data rates.

Rather than going to this menu to change the traffic channel number, you can also enter a new channel number in any of the measurement menus. This menu, however, allows you to change more than one parameter in one handoff; you can, for example, assign a new channel number in another frequency band and/or change the radio configuration and service option at the same time.



Function

By entering one or multiple new parameter values and pushing the **Execute** softkey, the mobile phone is directed to the new channel allocation. When the hard handoff is complete and the new channel allocation is in force, the bottom of the screen will indicate the new values.

This menu is accessible only while a call is established.

Setting up the hard handoff parameters

Changes of the parameters shown below take effect only after pushing the **Execute** softkey. Once the hard handoff has been completed successfully, the new parameter values will be displayed in the test parameters section at the bottom of the screen.

Band — This selection field allows you to change the ongoing call from the current frequency band (as indicated at the bottom of the screen) to another one. Only bands supported by the mobile phone will be available.

Channel — This entry field allows you to define a new carrier frequency for the traffic channel. See section "[Basic specifications of CDMA bands](#)" on page 95 for a list of valid channel numbers for each band.

Radio Configuration — With this selection field, you can change the radio configuration. The radio configurations and their meanings are explained in section "[Radio configurations, service options](#)" on page 97.

Service Option — The selection field provides access to the service option which determines the data rate and type of service (voice, data) for the call. The service options are listed in "[Radio configurations, service options](#)" on page 97.

System parameters

These menus allow to set a number of system parameters, to set test limits or to define a coupling loss curve.

To gain access to these menus, simply push the **Param...** softkey while the mobile phone is in idle state.

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

- ["Forward link" on page 74](#) – in this parameter menu, the physical parameters of the link to the mobile phone can be set.
- ["Network setup" on page 77](#) – allows to set the configuration data of the simulated base station.
- ["Traffic channel" on page 79](#) – this is the menu to define the traffic channel parameters such as the radio configuration, service option, closed-loop power control mechanism, as well as fundamental and supplemental channel parameters.
- ["Limits menus" on page 83](#) – The Limits softkey provides access to a number of menus for the various test limits. 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.
- ["Coupling loss" on page 89](#) – allows to compensate the losses of cables, splitters and other RF equipment. This is especially important when working with the universal antenna coupler.
- ["Auxiliary outputs" on page 94](#) – provides access to the setup of the CDMA-specific output connectors as far as they are user-configurable.

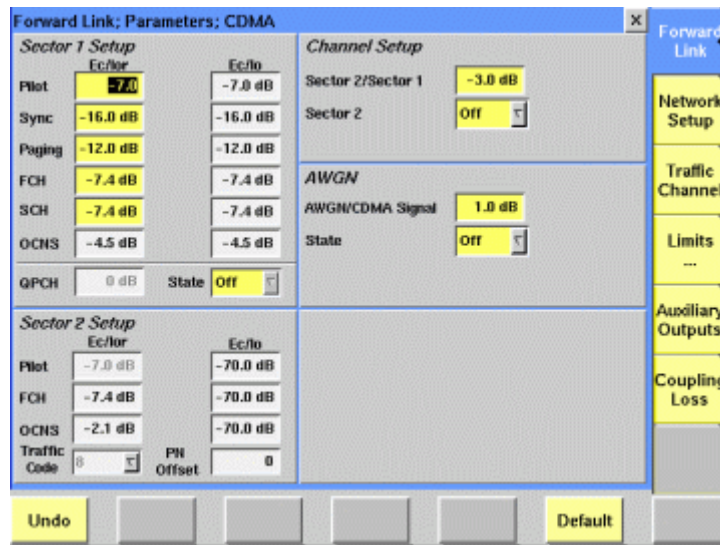
NOTE

Each of the parameter menus includes an **Undo** function. By pushing this softkey, all parameter values will return to their respective state before calling up the menu in question.

NOTE

Each of the parameter menus includes a **Default** softkey, setting the parameters on the respective screen to their default (factory) settings.

Forward link The forward link menu allows you to set a number of parameters characterizing the link to the mobile station.



Sector 1 setup In this section of the menu, you can enter the relative power of the channels for sector 1 of the simulated base station. The values are defined in terms of E_c/I_{or} . The calculated values of E_c/I_o , i.e. the energy relative to the sector 1 power, are displayed in the right-hand column.

Pilot The forward pilot channel is the unmodulated signal transmitted continuously by the base station to allow a mobile phone to acquire the timing and phase reference of the forward channel, as well as allowing signal strength comparisons between base stations.

Entry range: -32 to -5
Step size: 0.1
Default: -7.0

Sync The sync channel is an encoded, interleaved, spread, and modulated signal that is used by the mobile phone to acquire initial time synchronization.

Entry range: -32 to -5
Step size: 0.1
Default: -16.0

Paging The paging channel is a code channel used for the transmission of control information and pages.

Entry range: -32 to -5
Step size: 0.1
Default: -12.0

FCH The forward fundamental channel is a code channel used for the transmission of user information.

Entry range: -32 to -5
Step size: 0.1
Default: -11.6

| | |
|--------------|--|
| SCH | The forward supplementary channel (SCH) carries user information to a mobile phone. Up to two supplementary channels may exist for each forward traffic channel. The SCH is only available with radio configuration 3 or higher. Entry range: -32 to -5 Step size: 0.1 Default: -7.4 |
| OCNS | Orthogonally coded noise source. The power is not selectable but is calculated to give the full power level after adding all the various relative channel powers. |
| QPCH | |
| QPCH | Level of the quick paging channel (QPCH) relative to the sector 1 pilot channel. This entry field is not accessible when the QPCH is deselected (see State input field). Entry range: -5 to +2 Step size: 1 Default: 0 |
| State | This selection field turns the QPCH for sector 1 on or off. Valid entries: ON, OFF Default: OFF |

Sector 2 setup In this section of the menu, you can enter parameters of the second simulated sector, in particular the relative power of the channels for sector 2 of the simulated base station when sector 2 is turned on (see [Channel setup](#) section for switching sector 2 on). The power values are defined in terms of E_c/I_0 . The calculated values of E_c/I_0 , i.e. the energy relative to the sector 2 power, is displayed in the right-hand column.

The second sector is useful for two applications: It enables testing of the mobile's handoff capability (between two sectors of a cell); and it allows the mobile phone to perform power measurements on the sector 2 pilot channel.

The mobile measures and reports the pilot power for both sectors when any of the voice service options is used.

| | |
|--------------|---|
| Pilot | The forward pilot channel is the unmodulated signal transmitted continuously by the base station to allow a mobile phone to acquire the timing and phase reference of the forward channel, as well as allowing signal strength comparisons between base stations. Entry range: -32 to -5 Step size: 0.1 Default: -7.0 |
|--------------|---|

| | |
|---------------------|---|
| FCH | The forward fundamental channel is a code channel used for the transmission of user and signaling information. Entry range: -32 to -5 dB Step size: 0.1 dB Default: -7.4 dB |
| OCNS | Orthogonally coded noise source. The OCNS power (in dB) is not selectable but is calculated to give the balance power level after adding all the various relative channel powers. |
| Traffic Code | This selection field allows you to enter the (Walsh) code channel for the simulated forward traffic channel on sector 2. (Note that the Walsh code for sector 1 can be set up on the Traffic Channel parameters menu.) Possible entries: 8, 12, 16, 20, 24, 28, 36, 40, 44, 48, 52, 56, 60 Default: 8 |
| PN Offset | This is an information field indicating the PN Offset of the sector 2 signal. The PN Offset for sector 1 can be defined in the Network Setup menu. |

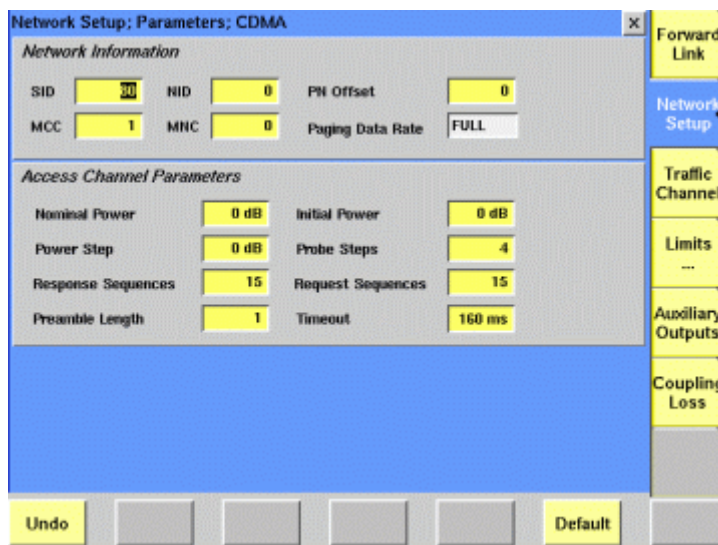
Channel setup

| | |
|----------------------------|--|
| Sector 2 / Sector 1 | This input field allows you to define the power level of the simulated second sector relative to that of the first sector, in dB. The power ratio is void if sector 2 is not enabled (see Sector 2 input field). Entry range: -12.0 to +3.0 Step size: 0.1 Default value: -3.0 |
| Sector 2 | This selection field allows you to switch the second sector on or off. Entry range: ON or OFF Default value: OFF |

AWGN

| | |
|---------------------------|---|
| AWGN / CDMA Signal | This input field allows you to define the level of the internally generated additive white gaussian noise relative to sector 1. Entry range: -10.0 to +5.0 Step size: 0.1 Default value: +1.0 |
| State | This selection field allows you to switch the AWGN generator on or off. Entry range: ON or OFF Default value: OFF |

Network setup The network setup menu allows to set the configuration data of the simulated base station.



Network information This section allows you to enter network- and base station-specific parameters that are broadcast by the base station simulated by the 4400.

- SID** This is the system identification (SID) code uniquely identifying a CDMA network. To avoid registration problems, select the home SID of the phone under test.
Entry range: 0 to 32767
Default: 1
- NID** The network identification (NID) number identifies the network.
Entry range: 0 to 65535
Default: 0
- PN Offset** Time offset to the PN sequence. Neighboring base stations use different PN offsets.
Entry range: 0 to 511
Default: 0
- MCC** The mobile country code identifies the country in which the network is located.
Entry range: 0 to 999
Default: 1
- MNC** The mobile network code (MNC) identifies a network within a country.
Entry range: 0 to 99
Default: 0
- Paging Data Rate** This output field indicates which paging data rate the network requires.

Access channel parameters

This section of the Network Setup menu allows you to define parameters that the mobile phone shall use while accessing the network. The parameters are broadcast by the simulated base station.

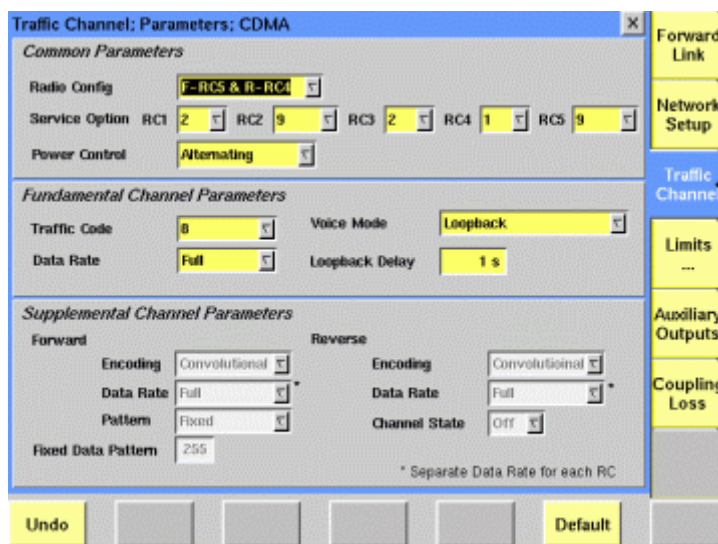
| | |
|---------------------------|---|
| Nominal Power | <p>This is the nominal power that the mobile phone is to transmit prior to applying power control. The nominal power is measured in dBm.</p> <p>Entry range: -24 to +7</p> <p>Default: 0</p> |
| Power Step | <p>The power step (in dB) is the step by which the mobile phone increases its output power when the previous access probe remained unanswered.</p> <p>Entry range: 0 to 7</p> <p>Default: 0</p> |
| Response Sequences | <p>This is the maximum number of response sequences that the mobile phone is allowed to use when accessing the network after being paged.</p> <p>Entry range: 1 to 15</p> <p>Default: 7</p> |
| Preamble Length | <p>The preamble length is the number of frames containing a preamble. The access channel preamble consists of frames of 96 zeros that are transmitted at a data rate of 4800 bps. The access channel preamble is transmitted to aid the base station in acquiring an access channel transmission.</p> <p>Entry range: 1 to 16</p> <p>Default: 1</p> |
| Initial Power | <p>The initial power is an offset (in dB) to the calculated transmit power of the mobile in open loop power control. The base station (or test equipment) commands the initial power on the paging channel.</p> <p>Entry range: -16 to 15</p> <p>Default: 0</p> |
| Probe Steps | <p>This is the number of probe steps in a sequence. If all access probes of the current access probe sequence remained unanswered, the mobile phone may try another sequence up to the maximum number of request sequences.</p> <p>Entry range: 0 to 16</p> <p>Default: 4</p> |
| Request Sequences | <p>This is the maximum number of request sequences that the mobile phone is allowed to use when accessing the network without being paged for it.</p> <p>Entry range: 1 to 15</p> <p>Default: 15</p> |

Timeout The acknowledgment timeout (ACC_TMO), in milliseconds, is the length of time mobile stations are to wait after the end of an access channel transmission before determining that the base station did not receive the transmission, in units of 80 ms. The mobile may actually extend the timeout by another 160 ms.
Entry range: 160 to 1360
Step size: 80
Default: 160

Traffic channel This menu affects the modulation, data rate and other important parameters for the traffic channel that the mobile phone will be assigned in a call setup.

NOTE

While on a traffic channel, any parameter change will take effect only when returning from the Parameters menus to one of the measurement menus.



Common parameters In this section, the radio configuration, the service option and the closed-loop power control mechanism can be defined.

Radio Config The radio configuration determines the modulation type and the allowable service that may be carried over the air interface (voice codec type or data rate). For a list of allowable radio configurations please refer to [Table 11 on page 99](#).

Service Option RC1 This selection field allows you to define the type of service if radio configuration 1 is selected. For a list of service options and their meanings see [Table 10 on page 98](#).

RC2 ...
RC5

These selection fields allow you to define the type of service for the respective radio configuration selected. For a list of service options and their meanings see [Table 10 on page 98](#). Note that the available service options depend on the actual radio configuration.

Power Control

The Power Control selection field defines the closed-loop power control mechanism applied on the forward link. Select between the following settings:

- ACTIVE lets the 4400 simulate a real base station by fine-tuning the mobile's transmit power to the optimum value so that the 4400 receives the signal at a suitable signal strength. The optimum level also depends on the 4400's power level; the calculation of the expected power is defined in the IS-2000 standard.
- ALTERNATING lets the simulated base station send an alternating power control bit pattern "01010101....".
- UP lets the 4400 send a power control bit pattern of "0000...", which commands the phone's power level to its highest value.
- DOWN lets the 4400 send a power control bit pattern of "1111...", which commands the phone's power level to its lowest value.

Note: Some of the measurement menus override this setting for the purpose of the test.

Fundamental channel parameters

Traffic Code

This selection field allows you to select the Walsh code for the traffic channel to be used on the forward link. The traffic code selected here does not affect high-speed data transmission; the latter will be sent with code group W3.

Possible entries: 8, 12, 16, 20, 24, 28, 36, 40, 44, 48, 52, 56, 60

Default: 8

Data Rate

This selection field allows you to select a data rate on the fundamental channel. Anything but the FULL data rate results in slotted operation (if not overruled by the type of test; see test descriptions). Possible entries: FULL | HALF | QUARTER | EIGHTH

Voice Mode This selection field defines the voice channel data to be sent by the 4400. Possible selections:

- Loopback. With this setting, the 4400 loops the voice data from the mobile phone back to it. The loopback delay can be defined with the Loopback Delay field.
- Generator-Speech. Sends a predefined speech pattern to the mobile's earpiece.
- Generator-400Hz Tone. Sends a predefined audio signal (a tone) to the mobile phone's earpiece.
- Generator-1kHz Tone. Sends a predefined audio signal (a tone) to the mobile phone's earpiece.
- Generator-Audio Sweep. Sends a predefined audio sweep signal (300 Hz to 3 kHz tone) to the mobile phone's earpiece.
- Silent – sends voice frames without any audible sound to the mobile phone.

Loopback Delay With this input field, you can define a loopback delay (in seconds) for the voice data. The delay time determines the time after which you hear your voice in the earpiece after talking into the microphone.
Range: 0 to 5
Default: 1

Supplemental channel parameters

These parameters are accessible only if any of the radio configurations 3, 4, or 5 and service option 32 are selected.

The supplementary channel parameters can be defined separately for the forward and the reverse link. A supplementary channel is transmitted when user data is transmitted at higher rates (above the nominal base rate for a channel).

An individual set of supplemental channel parameters can be entered for each of radio configurations 3, 4, and 5. For radio configurations 1 and 2, no supplemental channel is supported.

Forward

This section allows you to define parameters for the forward supplemental channel.

Encoding The Encoding selection field allows you to define the forward error correction (FEC) algorithm to be used. The two choices are Convolutional and Turbo.

Data Rate This input field allows you to select multiples of the base rate (DRATE) for the forward traffic channel. This results in the appropriate number of code channels to be transmitted. See [Table 12](#) for a translation of DRATE into data bit rates and available rates depending on the radio configuration. You can select one of the following values:

- FULL
- DRATE 2X
- DRATE 4X
- DRATE 8X
- DRATE 16X
- DRATE 32X

Note: A separate data rate can be selected for each radio configuration (RC). This means that the choice you make is applicable only for the currently selected RC. To define the rate for another RC, simply select the respective value in the Common Parameters section.

Pattern On the forward supplemental channel, the 4400 can send either a fixed data pattern or a pseudorandom bit sequence (PRBS).

Fixed Data Pattern The fixed data pattern is a repeated 8-bit sequence. You can enter the 8-bit value as a decimal number.

Entry range: 0 to 255

Default: 255

Reverse

This section allows you to define parameters for the reverse supplemental channel.

Encoding This selection field selects the forward error correction (FEC) algorithm to be used on the reverse link. You can choose between Convolutional and Turbo coding.

Data Rate This input field allows you to select multiples of the base rate (DRATE) for the reverse traffic channel. This results in the appropriate number of code channels to be transmitted. See [Table 13](#) for a translation of DRATE into data bit rates and available rates depending on the radio configuration. You can select one of the following values:

- FULL
- DRATE 2X
- DRATE 4X
- DRATE 8X
- DRATE 16X
- DRATE 32X

Note that the data pattern on the reverse channel is always PRBS.

Note: A separate data rate can be selected for each radio configuration (RC). This means that the choice you make is applicable only for the currently selected RC. To define the rate for another RC, simply select the respective value in the Common Parameters section.

Channel State The Channel State selection field defines whether the reverse supplemental channel is used or not. Select between ON and OFF.

Channel state off implies that the mobile does not need to transmit at higher data rates; consequently, code power tests are not possible in this mode.

Limits menus The 4400 can display 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 the [Limits menus](#). The Limits menus comprise:

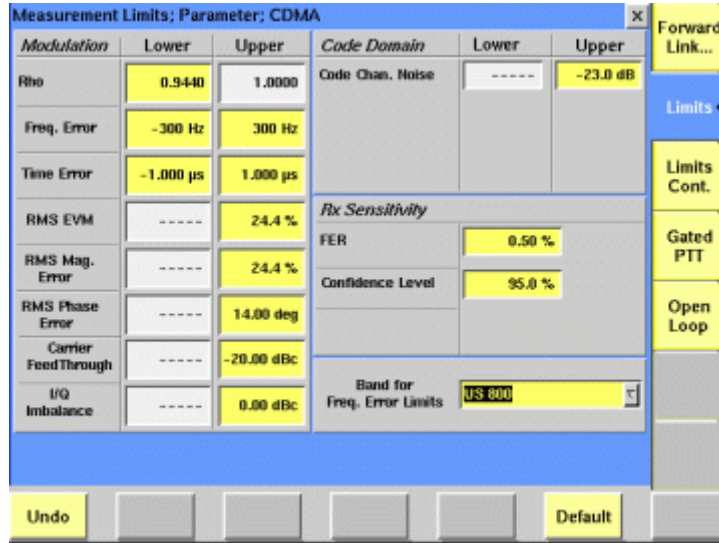
- Limits
- Limits Cont. (second limits menu)
- Gated PTT (power/time template)
- Open Loop (limits for the open loop power measurement)

How to read and enter the measurement limits

For most measurement parameters, 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.

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

Limits



Different sets of limits apply to the respective frequency band. By selecting another band, the stored limits for that band are loaded.

- Rho** The waveform quality factor rho is a measure of the modulation quality. The ideal value is 1; measured values are in the range 0 to 1.
Here you can enter the lower limit for measurements on the basic menu and the Rho/Freq. menu.
IS-98 section 10.2.3.2. specifies that rho must be greater than 0.944.
- Freq. Error** The upper and lower limits for the frequency error can be entered in Hertz. The results (with pass/fail verdict according to these limits) are shown on the basic menu and the Rho/Freq. menu.
- Time Error** The upper and lower time error limits are configurable in the range from -5 to +5 μ s. The resulting pass/fail verdict can be observed on the basic menu and on the Rho/Freq. menu.
- RMS EVM** The RMS error vector magnitude is a positive number in %; the ideal value is zero. For this reason, a lower limit is not configurable. The upper limit can be set in the range from 0 to 100. Measurement results are available on the Rho/Freq. measurement menu.
- RMS Mag. Error** Results of the RMS magnitude error measurement are positive numbers in %; the ideal value is zero. A lower limit is therefore not configurable. The upper limit is configurable in the range 0 to 100. Measurement results are available on the Rho/Freq. measurement menu.

| | |
|----------------------------|---|
| RMS Phase Error | Results of the RMS phase error measurement are positive numbers in degrees; the ideal value is zero. A lower limit is therefore not configurable. The upper limit is configurable in the range 0 to 45. Measurement results are available on the Rho/Freq. measurement menu. |
| Carrier Feedthrough | Results of the carrier feedthrough measurement are negative numbers in dBc; the ideal value is (-infinity). A lower limit is therefore not configurable. The upper limit is configurable in the range -99 to 0. Measurement results are available on the Rho/Freq. measurement menu. |
| I/Q Imbalance | Results of the I/Q imbalance measurement are negative numbers in dBc; the ideal value is (-infinity). A lower limit is therefore not configurable. The upper limit is configurable in the range -99 to 0. Measurement results are available on the Rho/Freq. measurement menu. |
| Code Chan. Noise | The ideal code channel noise is nonexistent (-infinity dB). A lower limit can therefore not be given. The upper limit (in dB) can be entered in the range from -70 to 0. Measurement results and the pass/fail verdict appear in the Code Domain menu. |
| FER | An upper limit for the frame error rate measurement (measured in % and displayed in the basic menu and in the FER menu) can be defined here. Valid entries are in the range 0 to 5.0. Note: This limit only applies to the FER measurements and not to the Reported FER measurements. |
| Confidence Level | This not a limit but a confidence level (in %) for the FER measurement. Valid entries are in the range 90 to 100. |
| Band | This selection field loads the limit parameters for a frequency band. The values for the previous band are saved. This way, different sets of limit values can be used for the various frequency bands. |

Limits continued

| Measurement Limits cont.: Parameters; CDMA | | | | | | Forward Link... |
|--|----------|-----------|-----------|----------|----------|-----------------|
| Power | Lower | Upper | ERP | Lower | Upper | Limits |
| Access Channel | -9.5 dB | 9.5 dB | Class I | 29.5 dBm | 36.0 dBm | Limits Cont. |
| Closed Loop Range | -24.0 dB | 24.0 dB | Class II | 25.5 dBm | 32.5 dBm | Gated PTT |
| Minimum Power | ----- | -50.0 dBm | Class III | 21.5 dBm | 28.5 dBm | Open Loop |
| Standby Power | ----- | -61.0 dBm | Class IV | -2.0 dBm | 36.0 dBm | |
| Band for ERP Limits | US 000 | | Class V | -2.0 dBm | 36.0 dBm | |

The limit values in this menu also depend on the frequency band. Ensure the correct band setting before checking or changing the limits!

Power limits

Access Channel

This limit refers to the access probe measurement in the **Power > Access Probe** menu. Valid entry values for the upper and lower limit (in dB) are in the range -9.5 to +9.5, relative to the expected power.

Closed Loop Range

The closed loop range limits refer to the upper and lower power measurement on the closed loop menu. The value in the Lower column determines the upper limit for the lower power, relative to the initial power value.

The value in the Upper column gives the lower limit for the upper power with respect to the initial power.

The defaults are -24 and +24 dB, respectively as defined in IS-98 section 4.4.4.3 (test 1a).

Minimum Power

The upper limit for the minimum power (in dBm) in the Min/Max power test menu can be entered here. The default is -50 dBm.

Standby Power

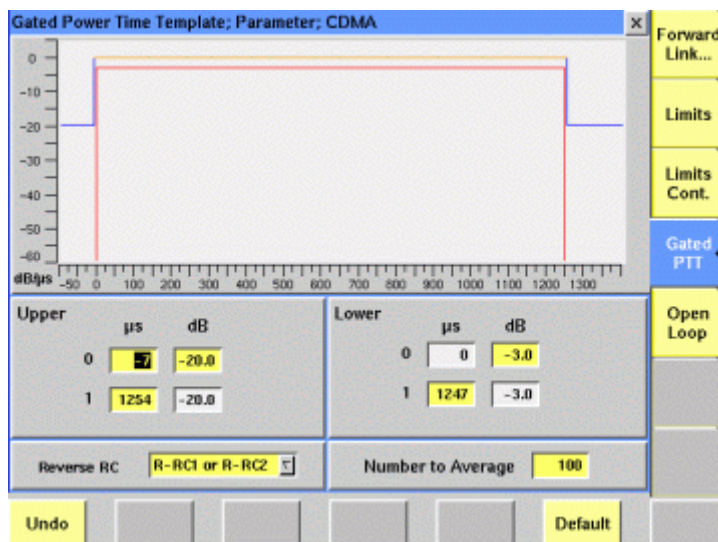
This limit refers to the standby power measurement in the **Power > Access Probe** menu. The upper limit represents the maximum power (in dBm) that the mobile may emit when in standby mode.

ERP limits

Class I through V

The ERP limits represent the maximum effective radiated power (ERP) that the mobile may emit at any time, and is different for each power class. These limits are applied during the Maximum power test. The lower and upper ERP limits for each power class can be entered in dBm.

Gated PTT



This menu allows you to define the limits for the gated power measurements. The transmission time differs depending on the active radio configuration; therefore, two different templates can be defined, one for radio configurations 1 and 2 and one for radio configurations 3 and 4.

For each template, three different sections can be defined:

- The upper limit before the active part of the transmission. The lower limit for this period cannot be defined because the lower the power outside the active part, the better.
- The lower limit for the active part of the transmission, relative to the average power of the gated-on duration. No upper limit is defined; the orange line at 0 dB is for reference purposes only.
- The upper limit after the active part of the transmission.

Upper – 0 – µs This input field lets you select the time up to which the upper limit before the active part applies. The time is entered in µs, relative to the start of the active part.

Upper – 0 – dB This input field defines the upper limit outside the active part of the transmission i.e. both before and after the active part. The limit is entered in dB.

Upper – 1 – µs This input field allows you to select the time from which on the upper limit after the active part applies. The time is given in µs, relative to the start of the active part.

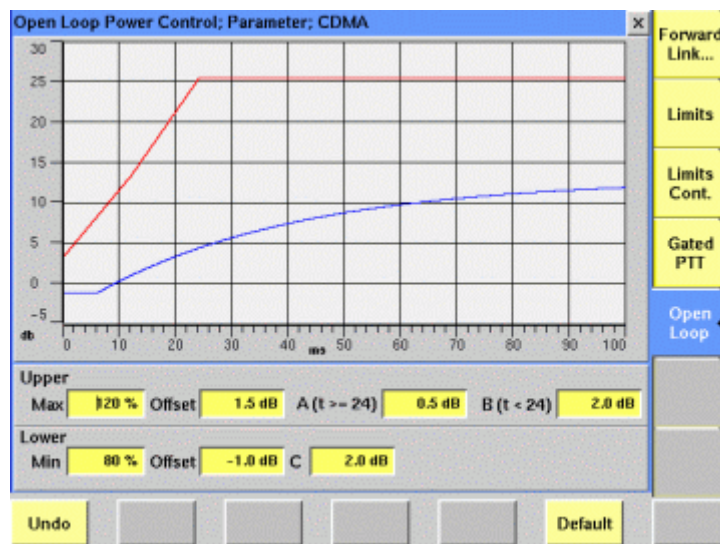
Upper – 1 – dB This field indicates the upper limit for the time after the active part. It can be changed only together with the upper limit before the active part.

Lower – 0 – µs This is the starting point of the lower limit for the active part of the transmission, given in µs relative to the starting point of the active part.

Lower – 0 – dB This is the lower limit during the active part of the transmission, in dB.

- Lower – 1 – μ s** This entry field describes the time when the end of the active part limit is reached. The time is entered in μ s, relative to the start of the active part.
- Lower – 1 – dB** This field repeats the lower limit for the active part of the transmission.
- Reverse RC** This selection field allows you to change the radio configuration for which you want to enter the gated PTT limits. The limits can be defined either for radio configurations 1 and 2 or for radio configurations 3 and 4.
- Number to Average** Enter the number of averages for a gated power measurement here. The valid range is from 1 to 200. An entry of 1 indicates that the measurements shall not be averaged.

Open loop



The upper and lower limits of the open loop power template depend on a number of factors which can be entered in this menu.

Upper limits

The upper limit for the Open Loop test is a disjoint limit, defined over two regions of time, namely for $t < 24$ ms and $t > 24$ ms. This limit is relative to the mobile's initial power at the start of the test procedure.

For $0 < t < 24$ ms, the upper limit is defined as:

$$\max [\text{Max} \cdot |\Delta P_{in}| \cdot (t/24), |\Delta P_{in}| \cdot (t/24) + B] + \text{Offset}$$

For $t \geq 24$ ms, the upper limit is defined as:

$$\max [\text{Max} \cdot |\Delta P_{in}|, |\Delta P_{in}| + A] + \text{Offset}$$

Several of these parameters used in setting the limits are user-definable.

| | |
|---------------------|---|
| Max | This parameter is expressed in percent of the power step size $ \Delta P_{in} $, e.g. 1.2 indicates 120% of the step size. The default for Max is 120% and the valid range is from 100 to 142. |
| Offset | This is an offset added at the end of each equation. The default value for it is 1.5 dB, it can be adjusted in the range from 0 to 8. |
| A (t>=24) | A is a factor added to $ \Delta P_{in} $ in the expression for $t \geq 24$ ms. The default value for A is 0.5 dB. |
| B (t<24) | B is a factor added to $ \Delta P_{in} $ in the expression for $t < 24$ ms. The default value for B is 2.0 dB. |

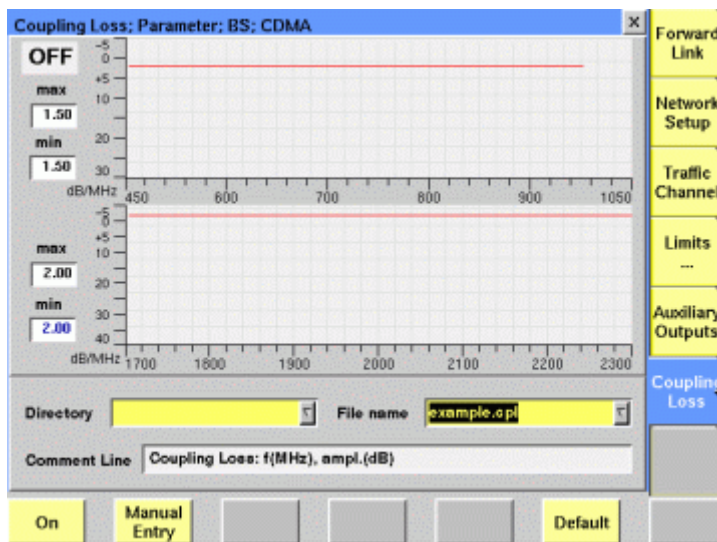
Lower limits

The lower limit for the Open Loop power test is also defined in IS-98D and given as:

$$\text{for } t \geq 0: \max [\text{Min} \cdot |\Delta P_{in}| \cdot [1 - e^{(1.25-t)/36}] - C, 0] - \text{Offset}$$

| | |
|---------------|---|
| Min | This parameter defaults to 0.8 or 80%. |
| Offset | This is the offset at the end of the equation and defaults to 1 dB. |
| C | This factor is used within the max[] expression for the lower limit and defaults to 2.0 dB. |

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. This attenuation associated with this loss typically varies with both the type of mobile being tested and the frequency. An example of a coupler being used is the Universal Antenna Coupler.



This menu allows compensation of 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 User-defined Attenuation menu.

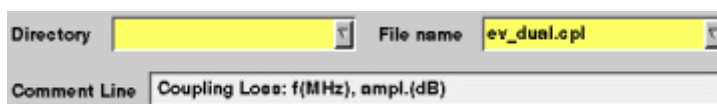
On the 4400, the contents of the currently selected *.cpl file will be displayed graphically for two frequency areas: The upper graph shows the coupling loss for the frequency range around 450 MHz and from 800 to 1000 MHz, while the lower graph shows the coupling loss for the frequency range 1700 to 2300 MHz.

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

The coupling loss parameters are stored in files; the file format is explained below.

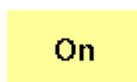
The file and directory structure allows you to store a set of parameters for each mobile and to locate the files for the phones from one vendor in one directory. The 4400 comes with a number of preinstalled directories for some of the larger manufacturers and a directory with parameter files for various "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



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 easily found at the beginning of the file list.



Push this softkey to activate the coupling loss compensation. 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 indication that the coupling loss compensation has been activated: 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 **ESCAPE** key.
- 4 Start RAPID! by first pushing the **TOOLS** 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 filename 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 **ESCAPE** key.
- 2 Start RAPID! by first pushing the **TOOLS** 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 around 450 MHz and from 800 to 1000 MHz and one for the frequency range from 1700 to 2300 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:

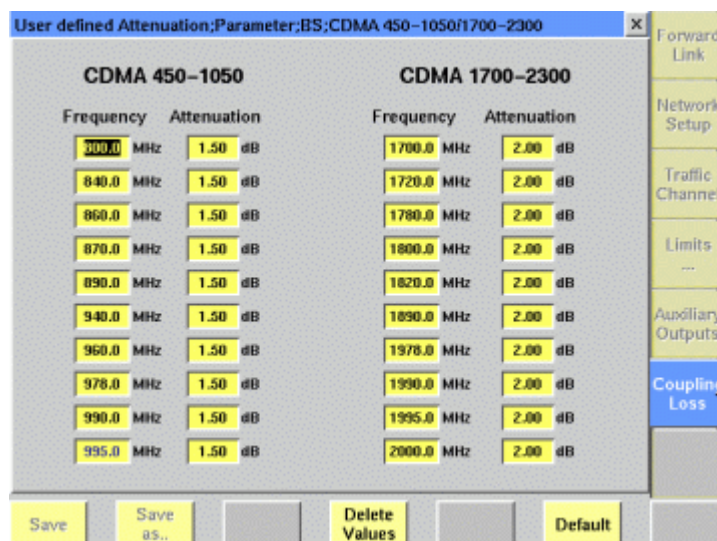
```
//Motorola P7389  
825.0,15.0  
1750.0,19.0
```

- 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 (450 to 1000, 1700 to 2300 MHz).



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

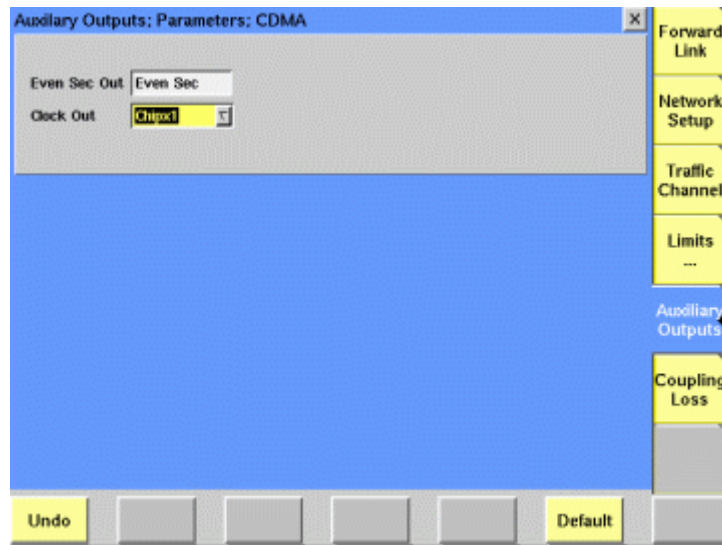
You can delete a row of entered values by placing the cursor in one of the three fields of that row and then pushing this softkey.

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 without selecting a file name, a previously existing file named `.manual.cpl` will be overwritten.

Auxiliary outputs



The Auxiliary Outputs menu allows you to configure hardware signals available at the connectors on the back of the instrument. Please see section [“Connectors on the RF section” on page 10](#) for a description of the physical interfaces.

Even Sec Out

The output of this connector is fixed to the even second clock signal. The pulse width is 50 ns with negative polarity.

Clock Out

This selection field determines the signal available at the AUX 6 BNC connector:

- `Chipx16` sets a clock of 16 times the chip rate, i.e. a clock signal of 19.6608 MHz will be available.
- `Chipx8` selects the eightfold chip rate, i.e. 9.8304 MHz.
- `Chipx1` selects the CDMA chip rate, i.e. 1.2288 MHz.
- `PCG` selects a clock signal recurring at the beginning of every power control group, i.e. at a frequency of 800 Hz.
- `Even Sec` selects the even second clock, i.e. at a frequency of 0.5 Hz, with a pulse width of 26.67 ms.
- `PN` selects the 26.67 ms cycle of the PN code. The pulse width is 200 ns.
- `Frame` generates a frame clock signal (every 20 ms i.e. at 50 Hz) at the beginning of each new (access channel, paging channel, traffic channel) frame.
- `S-Frame` generates a frame clock signal for the sync channel, recurring every 80 ms (12.5 Hz).

CDMA basics

In this section, you will find a short introduction to CDMA system parameters.

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

- [Basic specifications of CDMA bands](#)
- [Radio configurations, service options](#)
- [Test modes](#)
- [SCM coding](#)
- [MS power classes](#)

Basic specifications of CDMA bands

The tables below give the specifications for the various CDMA bands.

Table 5 CDMA frequency bands

| Band class | Band description | Reverse link (MS transmit) | Forward link (BS transmit) |
|------------|--------------------|----------------------------|----------------------------|
| 0 | U.S. cellular | 824 to 849 MHz | 869 to 894 MHz |
| 1 | North American PCS | 1850 to 1910 MHz | 1930 to 1990 MHz |
| 2 | TACS | 872 to 915 MHz | 917 to 960 MHz |
| 3 | JTACS | 887 to 925 MHz | 832 to 870 MHz |
| 4 | Korean PCS | 1850 to 1780 MHz | 1840 to 1870 MHz |
| 5 | NMT-450 | 411 to 484 MHz | 421 to 494 MHz |
| 6 | IMT-2000 | 1920 to 1980 MHz | 2110 to 2170 MHz |
| 7 | 700 MHz | 776 to 794 MHz | 746 to 764 MHz |
| 8 | 1800 MHz | 1710 to 1785 MHz | 1805 to 1880 MHz |
| 9 | 900 MHz | 880 to 915 MHz | 925 to 960 MHz |

Table 6 Channel arrangements

| Band class | Channel number | Center frequency, reverse link | Center frequency, forward link |
|------------|----------------|-------------------------------------|--------------------------------|
| 0 | 1 to 799 | $0.03 \cdot n + 825$ | $0.03 \cdot n + 870$ |
| | 991 to 1023 | $0.03 \cdot (n - 1023) + 825$ | $0.03 \cdot (n - 1023) + 870$ |
| 1 | 0 to 1199 | $0.05 \cdot n + 1850$ | $0.05 \cdot n + 1930$ |
| 2 | 0 to 1000 | $0.025 \cdot n + 889.9875$ | $0.025 \cdot n + 934.9875$ |
| | 1329 to 2047 | $0.025 \cdot (n - 1328) + 871.9875$ | $0.025 \cdot n + 916.9875$ |

Table 6 Channel arrangements

| Band class | Channel number | Center frequency, reverse link | Center frequency, forward link |
|------------|----------------|--------------------------------|--------------------------------|
| 3 | 1 to 799 | $0.0125 \cdot n + 915$ | $0.0125 \cdot n + 860$ |
| | 801 to 1039 | $0.0125 \cdot (n-800) + 898$ | $0.0125 \cdot (n-800) + 843$ |
| | 1041 to 1199 | $0.0125 \cdot (n-1040) + 887$ | $0.0125 \cdot (n-1040) + 832$ |
| | 1201 to 1600 | $0.0125 \cdot (n-1200) + 893$ | $0.0125 \cdot (n-1200) + 838$ |
| 4 | 0 to 599 | $0.05 \cdot n + 1750$ | $0.05 \cdot n + 1840$ |
| 5 | 1 to 300 | $0.025 \cdot (n-1) + 450$ | $0.025 \cdot (n-1) + 460$ |
| | 539 to 871 | $0.025 \cdot (n-512) + 411$ | $0.025 \cdot (n-512) + 421$ |
| | 1039 to 1473 | $0.025 \cdot (n-1024) + 451$ | $0.025 \cdot (n-1024) + 461$ |
| | 1792 to 2016 | $0.025 \cdot (n-1792) + 479$ | $0.025 \cdot (n-1792) + 489$ |
| 6 | 0 to 1199 | $0.05 \cdot n + 1920$ | $0.05 \cdot n + 2110$ |
| 7 | 0 to 359 | $0.05 \cdot n + 776$ | $0.05 \cdot n + 746$ |
| 8 | 0 to 1499 | $0.05 \cdot n + 1710$ | $0.05 \cdot n + 1805$ |
| 9 | 0 to 699 | $0.05 \cdot n + 880$ | $0.05 \cdot n + 925$ |

Table 7 Preferred channel numbers

| Band class | Block designator | Preferred channel numbers |
|------------|------------------|--|
| 0 | A | Subclass 0: 283, 691; subclass 1: 779, 738 |
| | B | Subclass 0: 384, 777; subclass 1: 486, 568 |
| 1 | A | 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275 |
| | B | 425, 450, 475, ..., 650, 675 |
| | C | 925, 950, 975, ..., 1150, 1175 |
| | D | 925, 950, 975, ..., 1150, 1175 |
| | E | 725, 750, 775 |
| | F | 825, 850, 875 |
| 2 | A | 79, 679, or 1365 |
| | B | 379, 947, or 1932 |
| 3 | A | Primary: 76; secondary: 872 |
| 4 | A | 25, 50, 75, 100, 125, 150, 175 |
| | B | 225, 250, 275, 300, 325, 350, 375 |
| | C | 425, 450, 475, 500, 525, 550, 575 |

Table 7 Preferred channel numbers

| Band class | Block designator | Preferred channel numbers |
|------------|------------------|-----------------------------------|
| 5 | A | 160, 210, 260 |
| | B | 120, 170, 220 |
| | C | 47, 97, 147 |
| | D | 573, 623, 673 |
| | E | 731, 781, 831 |
| | F | 1841, 1903, 1965 |
| | G | 1291, 1353, 1415 |
| | H | 1089, 1151, 1213 |
| 6 | | 25, 50, ..., 1150, 1175 |
| 7 | C | 45, 70, 95 |
| | D | 145, 170, 195, 220, 245, 270, 295 |
| 8 | | 25, 50, ..., 1450, 295 |
| 9 | | 25, 50, ..., 650, 675 |

Radio configurations, service options

The radio configuration specifies the type of data channel being used. The radio configurations can be defined for the forward and the reverse links as follows:

Table 8 cdmaOne/CDMA2000 reverse link radio configuration

| Radio configuration | IS-95 compatible | Data rates (kbs) | Base rate (kpbs) | Modulation type |
|---------------------|------------------|--|------------------|-----------------|
| RC 1 | x | 1.2, 2.4, 4.8, 9.6 | 9.6 | OQPSK |
| RC 2 | x | 1.8, 3.6, 7.2, 14.4 | 14.4 | OQPSK |
| RC 3 | | 1.2, 1.35, 1.5, 2.4, 2.7, 4.8, 9.6, 19.2, 38.4, 76.8, 153.6, 307.2 | 14.4 | HPSK |
| RC 4 | | 1.8, 3.6, 7.2, 14.4, 28.8, 57.6, 115.2, 230.4 | 14.4 | HPSK |

Table 9 cdmaOne/CDMA2000 forward link radio configuration

| Radio configuration | IS-95 compatible | Data rates (kbs) | Base rate (kbps) | Modulation type (traffic channel) |
|---------------------|------------------|--|------------------|-----------------------------------|
| RC 1 | x | 1.2, 2.4, 4.8, 9.6 | 9.6 | BPSK |
| RC 2 | x | 1.8, 3.6, 7.2, 14.4 | 14.4 | BPSK |
| RC 3 | | 1.2, 1.35, 1.5, 2.4, 2.7, 4.8, 9.6, 19.2, 38.4, 76.8, 153.6 | 9.6 | QPSK |
| RC 4 | | 1.2, 1.35, 1.5, 2.4, 2.7, 4.8, 9.6, 19.2, 38.4, 76.8, 153.6, 307.2 | 9.6 | QPSK |
| RC 5 | | 1.8, 3.6, 7.2, 14.4, 28.8, 57.6, 115.2, 230.4 | 14.4 | QPSK |

Table 10 Service options

| Service option | Type | Applicable radio configurations |
|----------------|--|-----------------------------------|
| 1 | Basic variable rate voice service (9.6 kbps) | F1-R1, F3-R3, F4-R3 |
| 2 | Mobile station loopback (9.6 kbps) | F1-R1, F3-R3, F4-R3 |
| 3 | EVRC (enhanced variable-rate voice service) (9.6 kbps) | F1-R1, F3-R3, F4-R3 |
| 9 | Mobile station loopback (14.4 kbps) | F2-R2, F3-R4 |
| 17 | High rate voice service (14.4 kbps) | F2-R2, F5-R4 |
| 32 | TDSO (test data service option) (9.6, 14.4 kbps) | F3-R3, F4-R3 |
| 55 | RC1/2/3/4/5 loopback (9.6, 14.4 kbps) | F1-R1, F2-R2, F3-R3, F4-R3, F5-R4 |
| 32768 | Basic variable rate voice service (14.4 kbps) | F2-R2, F5-R4 |

The following table lists the service options available for a radio configuration.

Table 11 Radio configurations and service options

| Radio configuration | Forward channel | Reverse channel | Supported service options |
|---------------------|-----------------|------------------|---------------------------|
| F1-R1 | BPSK, 9.6 kbps | OQPSK, 9.6 kbps | 1, 2, 3, 55 |
| F2-R2 | BPSK, 14.4 kbps | OQPSK, 14.4 kbps | 9, 17, 55, 32768 |
| F3-R3 | QPSK, 9.6 kbps | HPSK, 14.4 kbps | 1, 2, 3, 32, 55 |
| F4-R3 | QPSK, 9.6 kbps | HPSK, 14.4 kbps | 1, 2, 3, 32, 55 |
| F5-R4 | QPSK, 14.4 kbps | HPSK, 14.4 kbps | 9, 17, 55, 32768 |

Test modes

Voice testing Voice testing may involve either a qualitative check of the overall signal path, or measurements at defined audio frequencies and levels.

For a qualitative check, the audio signal from the microphone of the phone is transmitted to the 4400 over the air and returned to the mobile phone, so that the audio signal can be heard in the speaker of the phone after an adjustable delay. Alternatively, you can use one of the audio signals generated by the CDMA2000 System Option:

- speech (canned voice),
- a single audio tone at 400 or 1000 Hz,
- an audio sweep,
- or just silence.

These tests are possible without the Audio Option.

Audio measurements require the Audio Option to be installed. You can use either an audio signal generated by the audio generator, or one of the above signals.

Any of the voice tests require one of the following service options: 1, 3, 17, or 32768.

NOTE

CDMA2000 does not support service option 1. Use another service option instead.

Loopback testing In this mode, the 4400 sends pseudorandom data to the mobile phone, which in turn loops the signal back to the 4400. The data which the 4400 receives is utilized for Frame Error Rate (FER) testing.

In this mode, the mobile takes the data frames received from the base station (or 4400) and "loops" these frames back out the transmitter.

The loopback requires one of the service options 2, 9, or 55.

Data channel testing

Data channel testing requires service option 32.

For high-speed data channel testing with multiple code channels, one of the CDMA2000 radio configurations (3, 4, or 5) and TDSO (service option 32) need to be set. For high data rates, only two different Walsh code sets can be used in the forward link (F-SCH); the code group supported by the 4400 is W3.

The number of code channels can be defined separately for the forward and the reverse link by the data rate parameter (see **Parameters > Traffic Channel** menu). The usable range depends on the radio configuration and the encoding method as shown in [Table 12](#) for the forward link and [Table 13](#) for the reverse link.

Table 12 Data rates on the forward supplemental channel

| Radio configuration | Encoding | Data rates [kbps] |
|---------------------|---------------|---|
| RC3 | Convolutional | 9.6 (1X), 19.2 (2X), 38.4 (4X), 76.8 (8X), 153.6 (16X) |
| RC4 | Convolutional | 9.6 (1X), 19.2 (2X), 38.4 (4X), 76.8 (8X), 153.6 (16X) |
| RC5 | Convolutional | 14.4 (1X), 28.8 (2X), 57.6 (4X), 115.2 (8X), 230.4 (16X) |
| RC3 | Turbo | 19.2 (2X), 38.4 (4X), 76.8 (8X), 153.6 (16X) |
| RC4 | Turbo | 19.2 (2X), 38.4 (4X), 76.8 (8X), 153.6 (16X), 307.2 (32X) |

Table 13 Data rates on the reverse supplemental channel

| Radio configuration | Encoding | Data rates [kbps] |
|---------------------|---------------|--|
| RC3 | Convolutional | 9.6 (1X), 19.2 (2X), 38.4 (4X), 76.8 (8X), 153.6 (16X) |
| RC4 | Convolutional | 9.6 (1X), 19.2 (2X), 38.4 (4X), 76.8 (8X), 153.6 (16X), 307.2 (32X) |
| RC3 | Turbo | 19.2 (2X), 38.4 (4X), 76.8 (8X), 153.6 (16X), 307.2 (32X; only if forward link is using RC4) |
| RC4 | Turbo | 28.8 (2X), 57.6 (4X), 115.2 (8X) |

SCM coding

The 8-bit Station Class Mark (SCM) field reported by the mobile phone provides information about its capabilities. For band class 0 and 1 terminals, SCM is coded as follows:

Table 14 Station Class Mark coding for band class 0 and 1 terminals

| Function | Bit number(s) | Setting |
|---|---------------|--|
| Extended SCM indication | 7 | Band class 0: 0xxx xxxx Band class 1: 1xxx xxxx |
| Dual mode | 6 | CDMA only: x0xx xxxx Dual mode: x1xx xxxx |
| Slotted class | 5 | Nonslotted: xx0x xxxx Slotted: xx1x xxxx |
| IS-54 power class | 4 | Always 0 xxx0 xxxx |
| 25 MHz bandwidth | 3 | Always 1 xxxx 1xxx |
| Transmission | 2 | Continuous xxxx x0xx DTX xxxx x1xx |
| Power class for band class 0 analog operation | 1- 0 | Class I xxxx xx00 Class II xxxx xx01 Class III xxxx xx10 Reserved xxxx xx11 |

MS power classes

The power classes of the mobiles vary between the CDMA bands. The codes stand for different RF power levels in different CDMA bands. The table below gives an overview of the maximum output power levels for each power class and band.

| Power Class Code | 1 | 2 | 3 | 4 | 5 |
|--|----------------|----------------|----------------|----------------|---------------|
| Band class 0: U.S. Cellular | 31 - 38 dBm | 27 - 34 dBm | 23 - 30 dBm | - | - |
| Band class 1: North American PCS | 28 - 33 dBm | 23 - 30 dBm | 18 - 27 dBm | 13 - 24 dBm | 8 - 21 dBm |
| Band class 2: TACS band | - | 31 - 38 dBm | 27 - 34 dBm | 23 - 30 dBm | - |
| Band class 3: JTACS band | 31 - 38 dBm | 27 - 34 dBm | 23 - 30 dBm | - | - |
| Band class 4: Korean PCS band | 28 - 33 dBm | 23 - 30 dBm | 18 - 27 dBm | 13 - 24 dBm | 8 - 21 dBm |
| Band class 5: NMT-450 band | 33 - 40 dBm | 28 - 35 dBm | 23 - 30 dBm | 18 - 25 dBm | - |
| Band class 6: IMT-2000 band | 28 - 33 dBm | 23 - 30 dBm | 18 - 27 dBm | 13 - 24 dBm | 8 - 21 dBm |

| | | | | | |
|--|----------------|----------------|----------------|----------------|---------------|
| Band class 6: IMT-2000 band, Japan | 20 - 24 dBm | 20 - 24 dBm | 18 - 24 dBm | 13 - 24 dBm | 8 - 21 dBm |
| Band class 7: 700 MHz band | 27 - 34 dBm | 23 - 30 dBm | - | - | - |
| Band class 8: 1800 MHz band | 28 - 33 dBm | 23 - 30 dBm | 18 - 27 dBm | 13 - 24 dBm | 8 - 21 dBm |
| Band class 9: 900 MHz band | 31 - 38 dBm | 27 - 34 dBm | 23 - 30 dBm | - | - |

CDMA Nonsignaling Mode

3

This chapter provides task-based instructions for using the nonsignaling mode (Generator/Analyzer) within the 4464 CDMA2000 System Option. Topics discussed in this chapter are as follows:

- ["Overview" on page 104](#)
- ["RF generator" on page 105](#)
- ["RF analyzer for rho and frequency" on page 107](#)
- ["RF analyzer for power" on page 107](#)
- ["RF analyzer for gated power" on page 109](#)
- ["RF analyzer for code domain power" on page 110](#)
- ["Common elements of the RF analyzer menus" on page 112](#)
- ["Parameter menus" on page 113](#)

Overview

The generator/analyzer mode of the 4400 provides test and measurement tools for asynchronous measurements; these are measurements without requiring a call setup.

This section consists of the following parts:

- **RF generator** – This section describes the menu of the RF signal generator.
- **RF analyzer for rho and frequency** – This menu provides measurement results for the modulation quality.
- **RF analyzer for power** – The overall power of the signal transmitted by the mobile phone is measured and displayed.
- **RF analyzer for code domain power** – This menu provides displays of the power versus time when the mobile phone transmits a gated signal.
- Signals using multiple code channels are analyzed and displayed in this menu.
- **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 (CDMA).
- 3 Push the **Gen/Ana** softkey.



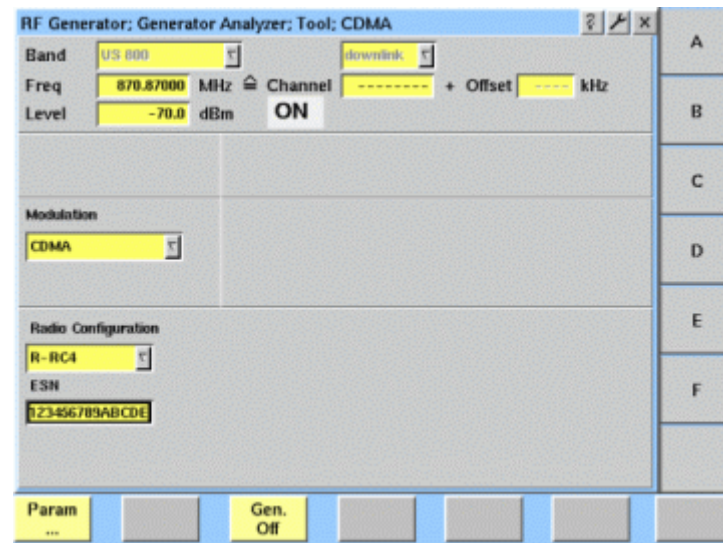
RF generator

To gain access to this menu, push the [RF Gen] marker tab while working with the generator/analyzer.

This menu allows you to set the parameters of the RF generator.

The RF generator can be set to generate radio signals at a defined carrier frequency.

Modulation can be BPSK or QPSK – or the signals can be totally unmodulated in order to stimulate pure RF circuitry.



What it does

The 4400 can be set up to generate a typical base station signal but without supporting any exchange of signaling messages. The generator can be used to test the receiver of the mobile phone in a special test (engineering) mode. It can also be used to provide a frequency and timing reference which the phone may require before transmitting any signal in test mode.

The channels transmitted by the 4400 generator are pilot channel, sync channel, paging channel, and orthogonal channel noise source (OCNS) channels. The respective level of each channel can be set up in the Parameter menus.

Parameters

Band – Selection field to set the frequency band.

Selections possible: US 800 | US PCS | TACS | JTACS | Korean PCS | NMT-450 | IMT-2000 | 900 MHz | 1800 MHz.

NOTE

To gain access to the Band and uplink/downlink entry fields, enter a channel number on the Channel entry field.

For details regarding CDMA frequency ranges, check with section ["Basic specifications of CDMA bands"](#) on page 95.

uplink/downlink — Selection field to specify either the uplink (reverse link) or the downlink (forward link) frequency range.
Selections possible: `uplink` or `downlink`.

Freq — Entry field to set the output frequency of the RF generator.
Entry ranges: 430 to 500 MHz, 800 to 1000 MHz and 1700 to 2300 MHz in increments of 10 Hz.

Channel — Entry field for the CDMA channel number. If you enter a new value in this field, the Frequency field will be recalculated based on the Band and uplink/downlink input fields provided that the channel number exists in this band.
Entry range: 0 to 2047

Offset — Entry field for the offset from the channel frequency specified.
Entry range: -99.99 to 99.99 kHz in increments of 10 Hz.

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 0.0 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 Ω .

ON/OFF

Display field for the current output status of the RF generator. When ON is displayed, RF signals are output; when OFF is displayed, the RF generator is silent.

To toggle between the two states, use the **Gen. Off** or **Gen. On** softkey, respectively.

Modulation — Selection field to turn the modulation on or off.
Selections possible: `CDMA` or `none`. If set to `CDMA`, the 4400 will transmit typical control channels using BPSK modulation (i.e. no traffic channel).

NOTE

Setting "none" means that the RF output signal generated by the 4400 will be unmodulated, i.e. a pure carrier sine wave.

Radio Configuration — Selection field for the received radio configuration. This parameter is used to correctly demodulate and measure the mobile phone's signal in the RF analyzer.
Selections possible: `R-RC1` | `R-RC2` | `R-RC3` | `R-RC4`

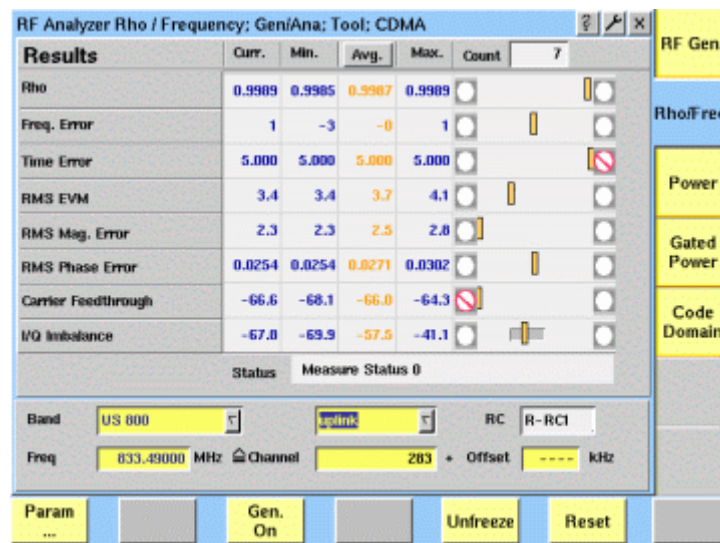
ESN — Electronic Serial Number of the mobile phone under test. This entry is required for the analysis (demodulation/decoding) of the phone's transmit signal when operating in `R-RC3` and `R-RC4`. The serial number can be entered as a hexadecimal number (see ["Entering values on hexadecimal entry fields" on page 24](#)).

RF analyzer for rho and frequency

To gain access to this menu, push the [Analyzer Rho/Freq.] marker tab while working with the generator/analyzer.
This menu displays the results of the modulation 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 CDMA-modulated and using the reverse radio configuration set on the RF Gen. menu.



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 CDMA test menu.

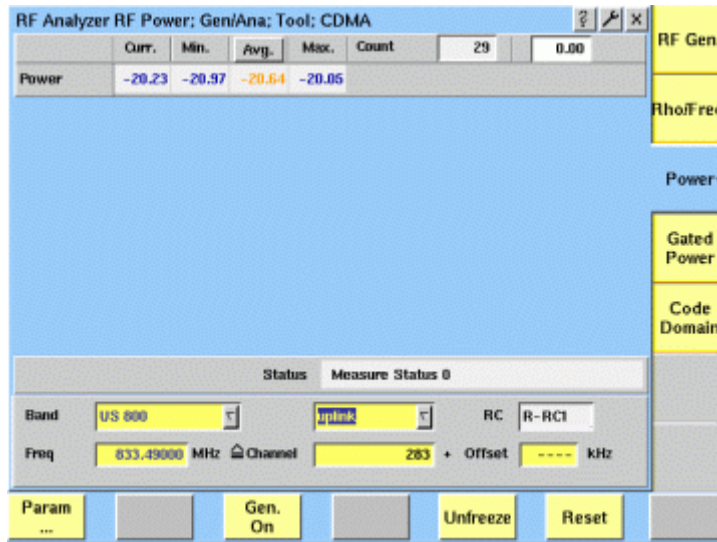
The frequency parameters and the softkeys are explained in section ["Common elements of the RF analyzer menus"](#) on page 112.

RF analyzer for power

To gain access to this menu, push the [Power] marker tab while working in generator/analyzer mode.
This menu displays the results of the power measurements of the RF signal received at the RF in/out connector while in nonsignaling mode.

NOTE

The analyzer will measure the signal and display the results assuming that the signal is CDMA-modulated and using the reverse radio configuration set on the RF Gen. menu.



The RF analyzer for power 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.

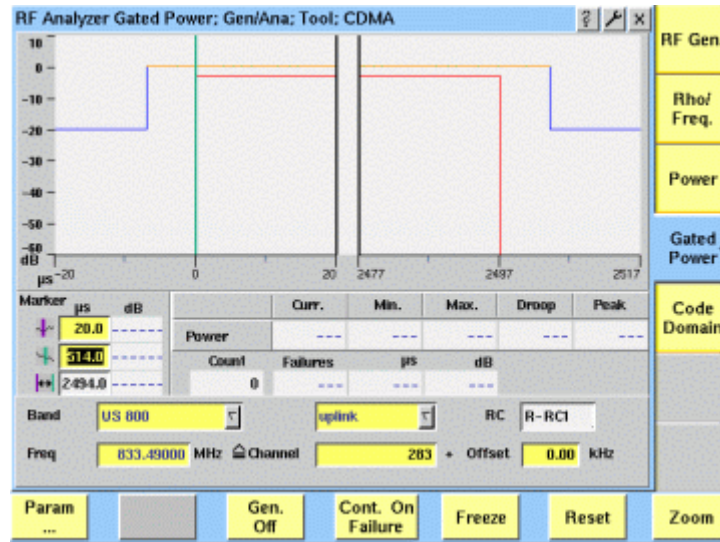
There are entry fields available for these parameters in the lower area of the menu, just above the softkeys.

The measurement results and the display markers are similar to the corresponding areas and fields of the standard CDMA test menu.

The frequency parameters and the softkeys are explained in section ["Common elements of the RF analyzer menus"](#) on page 112.

RF analyzer for gated power

From any of the CDMA Generator/Analyzer menus, push **Gated** to see the gated power menu.



Function

In the gated power test menu, the 4400 assumes that the mobile phone transmits at a substrate so that the phone only transmits every second, fourth or eighth gated-on period. This allows the user to examine the gated signal (power vs. time) against the specified template. The graphical display shows the gated power versus time, averaged over a number of transmissions (bursts). The number of measurements over which the 4400 averages the results can be defined on the **Parameters > Limits > Gated PTT** menu.

Note that the transmission time and the template depend on the selected radio configuration. The template for gating can be defined for the different radio configurations in the **Parameters > Limits > Gated PTT** menu.

How to read the gated power test results display

The graphical display shows the mobile's transmit power versus time for one gated-on period (burst). The horizontal scale is microseconds and the vertical scale is dB relative to the average power for the gated-on period. The upper and lower limits are indicated by straight lines in red and blue. The limits can be changed by pushing **Param > Limits > Gated PTT**.

The Power result field indicates the RF power (in dBm) over a gated-on period (burst). The statistical evaluation of the measurement results is displayed as current, minimum, average and maximum values.

Viewing certain details of the gated power

The initial graphical representation of the gated power shows the power versus time at both edges of the burst. You can view other parts of the burst by pushing the **Zoom** softkey.

Upon the first push on the **Zoom** softkey, the power over the whole gated-on period is displayed. The next pushes on the **Zoom** softkey result in the display of the rising and falling edges, respectively. Another push on this button will return to the initial gated power display.

Softkeys in the gated power menu

The softkeys provide different functions to control the functions and the result output. The standard softkeys are explained in "Softkeys of the RF analyzer" on page 113. Additional softkeys may be available.

Cont. On Failure

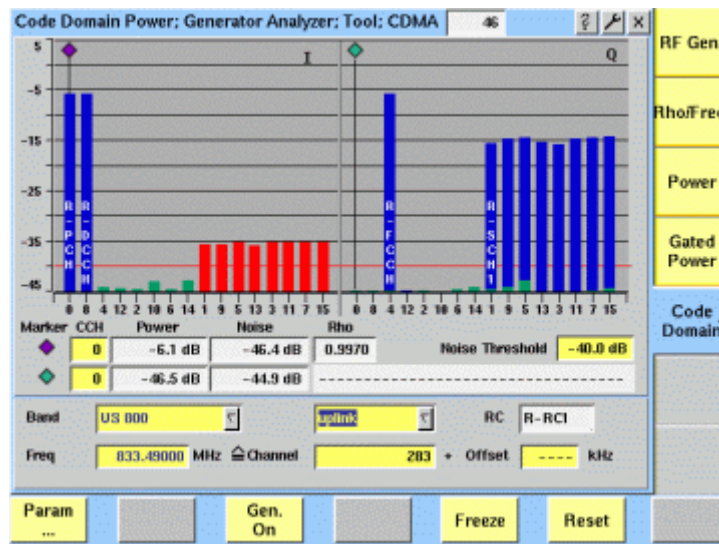
This softkey can be used to determine the behavior of the 4400 when the results are out of limit. A push on the **Cont. On Failure** softkey makes the 4400 continue measuring and updating the display when the results indicate a failure in the mobile phone. The softkey description changes to **Freeze On Failure**; if you select this setting, the results are not updated after an error occurred so that you can view the error.

Zoom

The function of the **Zoom** softkey is described above.

RF analyzer for code domain power

From any of the CDMA Generator/Analyzer menus, push **Code Domain** to see the code domain power menu.



Function When the mobile phone transmits high-speed data using multiple code channels, the Code Domain test menu shows the transmit power for each code channel, relative to absolute power for the complete transmit signal. It is assumed that the phone is brought into a state of transmitting multiple code channels without the help of a call setup.

How to read the code domain test results display

The live display shows bars indicating the power. The measurements are separated by both code channels and I/Q proportions. Different colors are used to mark active channels and noise as follows:

- Active channels are shown in blue.
- The code channel noise is colored green. The code channel noise is measured by subtracting the reference signal (i.e., the "ideal" signal) from the received signal, obtaining an error signal, and then despreading this error signal with all of the W_n^{16} codes. This is done for both I and Q components of the error signal.
- If the level of an inactive channel code exceeds the noise threshold, it is colored red.

One marker for each phase (I, Q) is available to read out both code channel power and code channel noise.

Code domain power parameters

Section ["Entry fields of the RF analyzer" on page 112](#) details the parameters for setting the measurement frequency.

The parameter significant for this menu is Noise Threshold. This value is used as a limit for unacceptably high noise levels. The threshold is indicated as a red horizontal line on the live display; any channel with a noise power exceeding this threshold is indicated by a red bar.

The softkey functions are explained in section ["Softkeys of the RF analyzer" on page 113](#).

Common elements of the RF analyzer menus

Entry fields of the RF analyzer

The radio frequency to measure on is set in this menu area.



Band — Selection field to set the frequency range that associates channel numbers with carrier frequencies.

Selections possible: US 800 | US PCS | TACS | JTACS | Korean PCS | NMT-450 | IMT-2000 | 900 MHz | 1800 MHz.

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, first enter a frequency in the Freq entry field that corresponds to a valid CDMA channel number for the selected band.

NOTE

For details regarding CDMA frequency ranges, check with section ["Basic specifications of CDMA bands"](#) on page 95.

RC — Display field indicating the selected radio configuration for the reverse channel. This is important for correct measurements of the transmitter quality parameters. The radio configuration can be set on the RF Gen. menu.

Freq — Entry field to set the measurement frequency for the RF analyzer. Entry range: 430 to 450 MHz, 800 to 1000 MHz and 1700 to 2300 MHz in increments of 10 Hz.

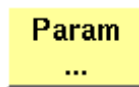
Channel — Entry field for the CDMA channel number. The entry range depends on the selected frequency band.

Offset — Entry field for the offset from the channel frequency specified. Entry range: -99.99 to 99.99 kHz in increments of 10 Hz.

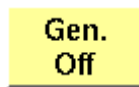
Avg — Click on this field or push the **Enter** key on this field to open a dialog box. The box allows you to define the number of last measurements results over which to average.

Select **OK** to confirm the current value or **Cancel** to return to the setting that was in use before opening this dialog box.

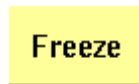
Softkeys of the RF analyzer

A yellow rectangular softkey with the text "Param" in bold and "..." below it.

This softkey leads you to the parameter screens where you can set up coupling loss factors and limits.

A yellow rectangular softkey with the text "Gen." in bold and "Off" below it.

A push on this softkey switches the [RF generator](#) either on or off.

A yellow rectangular softkey with the text "Freeze" in bold.

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.

A yellow rectangular softkey with the text "Reset" in bold.

A push on this softkey resets the statistic evaluation of the measurement results. The measurement counter will be reset as well.

Parameter menus

The Parameter menus allow you to access additional parameters relevant in both nonsignaling and in call mode at a more detailed level, which need not be changed in everyday use. To access the Parameter menus, press the **Param...** softkey on any of the generator/analyzer menu and then press the appropriate marker tab.

More details about the parameter menus can be found in section "[System parameters](#)" on page 73. Note that some parameters may not be applicable to the nonsignaling mode, e.g. open loop limits or limits for different ERP classes. The level specified for the fundamental and supplemental channels in the forward link parameters screen is applied to the predefined OCNS channels.

AMPS Call Mode

4

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

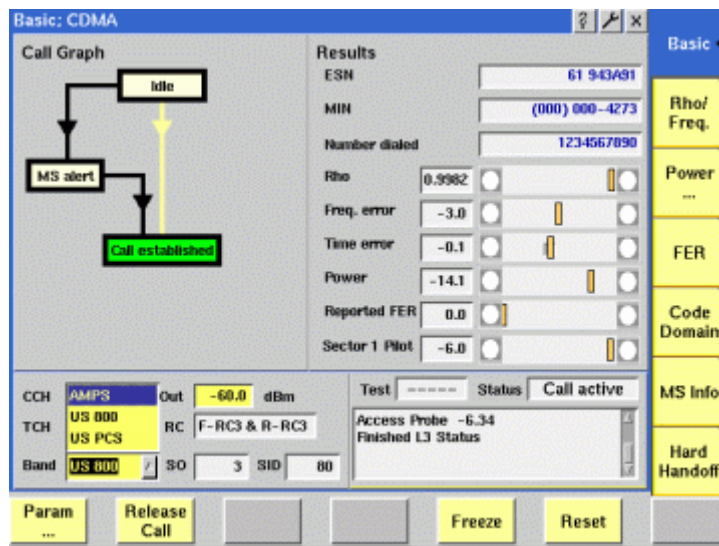
- ["Overview" on page 116](#)
- ["Common elements of the call mode menus" on page 117](#)
- ["Testing the phone on AMPS channels" on page 120](#)
- ["Parameters" on page 125](#)

Overview

This section covers information on AMPS measurements in synchronous (call) mode. These are available when the 4464 CDMA2000 System Option is installed. For information on AMPS measurements, see ["Testing the phone on AMPS channels" on page 120](#). See also ["Common elements of the call mode menus" on page 117](#) for more information. Additional test parameters and limits are explained in section ["Parameters" on page 125](#).

The CDMA2000 System Option supports transmitter, receiver and signaling tone measurements on AMPS voice channels for CDMA phones with analog AMPS capability. AMPS measurements are accessible after a handoff from a CDMA to an AMPS channel; the CDMA band must be either US 800 or US PCS. The call can also be handed over to another AMPS channel to enable measurements on different carrier frequencies of the cellular band (between 800 and 900 MHz). Please note that the AMPS and CDMA standards only support handoffs from CDMA to AMPS, not vice versa.

You will be able to access the handoff to AMPS after selecting CDMA on the Welcome menu, pressing the **Calls & Meas.** softkey and setting up a mobile or base station-originated call in either the US 800 or the US PCS band. Simply select AMPS as the new band and the 4400 will handoff to an AMPS channel.



Common elements of the call mode menus

Test parameters

The screenshot shows a menu with four entries, each with a yellow input field: VCH (29), MAC (2), SAT (1), and RF Out (-60.0 dBm).

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

VCH — Channel number for the voice channel. The entry can be changed while on an AMPS channel, leading to a handoff to another AMPS voice channel. This implies that the AMPS channel cannot be set up in idle mode prior to the call; the 4400 uses the channel number that was active during the last AMPS connection.

The valid input range is from 1 to 799 and from 991 to 1023. The channels are in the range from 869 to 894 MHz (forward link) and 824 to 849 MHz (reverse link).

MAC — Mobile attenuation code that the base station transmits to the mobile phone to control the phone's transmit power level during a call. Valid range is from 0 to 7, corresponding to a nominal power level as follows (note that nominal power levels exceeding the mobile's capabilities given by its power class, result in the highest possible power level):

Table 15 Mobile attenuation codes (MAC) and nominal power levels

| MAC | Nominal power level |
|-----|---------------------|
| 0 | 36 dBm |
| 1 | 32 dBm |
| 2 | 28 dBm |
| 3 | 24 dBm |
| 4 | 20 dBm |
| 5 | 16 dBm |
| 6 | 12 dBm |
| 7 | 8 dBm |

SAT — Supervisory audio tone. One of three tones (5970, 6000, 6030 Hz) transmitted by the cell base station and transponded by the mobile phone. Used to evaluate the complete radio path, both forward and reverse voice channels. The SAT received by the mobile unit is actually regenerated by the mobile unit with the same tone associated with the actual received SAT.

The valid input range is from 0 to 2, corresponding to the above values.

RF Out — RF output level of the 4400's signal, or the input level at the mobile phone if the coupling loss is selected correctly.
The valid range is from -120 to -5 dBm.

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.

| | |
|----------------------------|---|
| 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. |
| ----- (six hyphens) | This indicates that the call setup has not been started yet or that there has been no communication between the 4400 and the mobile so far. |

The large display field below provides more details and a short signaling history. On some of the menus, you can even scroll back within the history using the **PgUP**, **PgDN**, **UP** and **DN** keys.

Statistical test evaluation

| | Curr. | Min. | Avg. | Max. | Count | 24 |
|-------|--------|--------|--------|--------|--------------------------|--------------------------|
| Upper | 25.49 | 24.52 | 26.18 | 28.23 | <input type="checkbox"/> | <input type="checkbox"/> |
| Lower | -50.83 | -51.16 | -49.90 | -46.71 | <input type="checkbox"/> | <input type="checkbox"/> |

The following information can be found in the results table:

- Count** On some of the test menus, the 4400 provides you with statistical evaluations of the measured values. Those statistical 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.
- Curr.** Current measurement value. The last measurement obtained without any statistical evaluation.
- Min.** The absolute minimum of all measurements obtained.
- Avg.** The average value calculated using a block of measurement values. See below for more information about averaging.
- Max.** The absolute maximum of all measurements obtained.

NOTE

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

The average value on the results screens is determined by the last N measurements. The number N can be selected with the "Number of samples to average" input box which is available on all of the measurement menus; to access this field, push **ENTER** on the Avg. input field or click on it with the mouse. The following input box pops up:

Number of samples to average:

You can enter any number from 1 to 100. Using the **RIGHT** cursor key, move the cursor to the **OK** button and push **ENTER**; alternatively, use the mouse to click on the **OK** button.

Softkeys The 4400 displays the following softkeys in the AMPS call mode menus.

Param
...

Calls the Limit menu on-screen. With the help of this menu, you may view and change the test limits.

Release Call

A push on this softkey terminates the call. When the phone is back to idle mode, the CDMA basic menu in idle mode appears.

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.

Testing the phone on AMPS channels

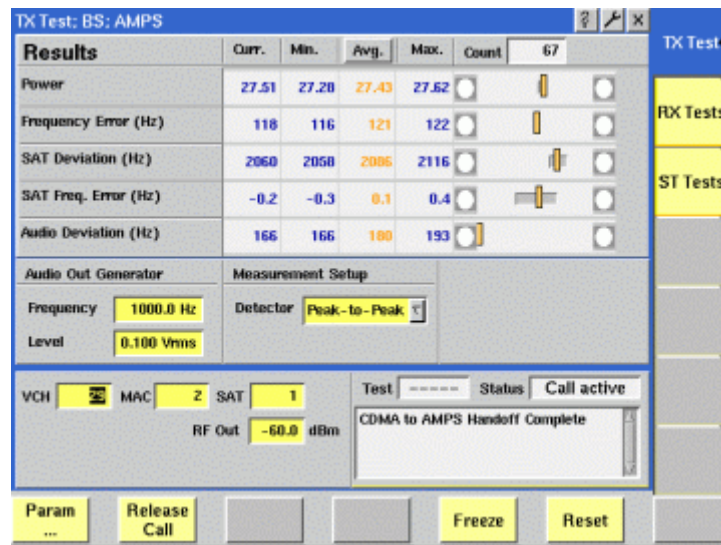
A few buttons and menus suffice to test the mobile phone and ensure its quality. The following sections will help you set up the 4400 for AMPS tests: Section ["Crash course" on page 4](#) gives a short guideline on how to connect a mobile and perform tests.

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.

The structure of this section reflects the different test menus available with the marker tabs:

- [TX Tests menu](#)
- [RX Tests menu](#)
- [ST Tests](#)

TX Tests menu This is the default menu appearing after a handoff from a CDMA to an AMPS channel. The menu allows you to view the transmitter parameters measured by the 4400.



What it does The TX Tests menu provides you with

- the transmit power
- the most important test results of the mobile's transmitter.

Transmitter measurements are based on three signals: The overall signal (affecting power level and frequency error), SAT (supervisory audio tone, affecting SAT deviation and SAT frequency error), and a user-defined audio tone (affecting audio deviation). The SAT signal is always present, so it is easy to test. The audio tone can be generated in the 4400 and provided to the mobile phone for these measurements.

Results display The test results display gives an overview of the mobile's transmitter characteristics.

NOTE

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

Test results display:

| | | | | |
|-----------------------------|-------|-------|-------|-------|
| Power | 27.51 | 27.28 | 27.43 | 27.62 |
| Frequency Error (Hz) | 118 | 116 | 121 | 122 |
| SAT Deviation (Hz) | 2060 | 2058 | 2086 | 2116 |
| SAT Freq. Error (Hz) | -0.2 | -0.3 | 0.1 | 0.4 |
| Audio Deviation (Hz) | 166 | 166 | 180 | 193 |

How to read the basic test results display

| | |
|------------------------|---|
| Power | This row indicates the power measurement, displayed as the mean output power of the mobile station's transmitter. The limits for this test are set on the measurement limits menu. |
| Frequency Error | Here, the 4400 displays the results of the statistical evaluation of the frequency error measurements. The result is given in Hertz. |
| SAT Deviation | This row shows the deviation of the SAT (supervisory audio tone). The SAT carrier is modulated with a nominal deviation of 2 kHz and the allowable limit is ± 0.2 kHz off the nominal value. |
| SAT Freq. Error | The SAT frequency error describes the difference between the measured and the nominal SAT carrier frequency. The nominal value depends on the SAT setting; see section "Test parameters" on page 117. The error must be close to zero. |
| Audio Deviation | This measurement indicates the frequency deviation affected by the transmitted audio (or DTMF) signal. The deviation should rise with the audio level but not exceed 12 kHz. The 4400 can provide an audio tone with defined frequency and level at the AF out connector, which can then be fed into an audio input at the mobile phone. See below for more details. |

Selecting measurement-specific parameters

In addition to the standard input parameters for the AMPS menus (see section "Test parameters" on page 117), the TX Tests menu allows to affect the following parameters:

AF out Generator: Frequency — This input field allows to enter the audio frequency that the 4400 should transmit at the **AF OUT** connector. The signal can be fed into an audio input at the mobile phone so that the phone has a signal to transmit over the RF. Another important parameter is the level (see below) at

which the 4400 transmits the signal. Note that the AF generator is available only if the 4470 Audio Option is installed.

Range: 20 to 20,000 Hz

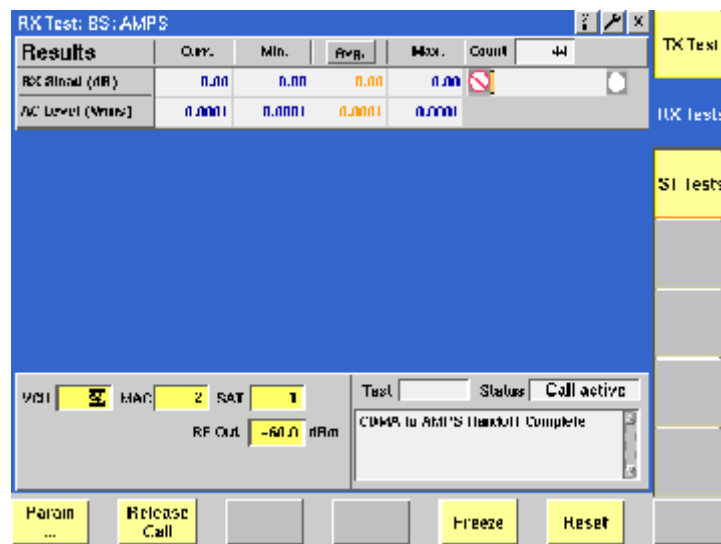
AF out Generator: Level — This input field directly affects the voltage level at which the 4400 transmits an audio signal at the **AF OUT** connector. See the frequency parameter for more details. Note that the AF generator is available if the 4470 Audio Option is installed.

Range: 0 to 4 V_{rms} in 0.001 V steps

Measurement Setup: Detector — This setting affects how frequency deviation is measured. The 4400 can either measure the positive peak (Peak+), the negative peak (Peak-) or the difference between the positive and negative peak (Peak-to-Peak).

RX Tests menu

This test menu gives you relevant parameters required to assess the quality of the mobile receiver. During the AMPS call, press the [RX Tests] marker tab to enter this menu.



NOTE

The receiver measurements are only supported with the 4470 Audio Option installed.

NOTE

If the CDMA call was started in loopback mode (service options 2, 9 or 55), the mobile phone may not receive the audio tone. As a workaround, perform an ST test first; the RX tests will work after the ST tests.

What it does

As long as this menu is active, the 4400 sends an audio signal over the voice channel so that the receiver quality can be assessed. The audio signal is a 1 kHz tone transmitted with 8 kHz deviation. The audio signal starts when the menu is selected and disappears when another menu is selected. Assessing the quality of the receiver is possible by analyzing the audio signal received in the mobile

phone. The received signal can be fed into the **AF IN** connector 4400 either with an audio cable (if the phone provides an appropriate connector) or with a microphone picking up the signal from the speaker of the phone.

The quality of the audio signal is affected by the level at which the phone receives the radio frequency signal, so the 4400 output power (RF out) is an important parameter for the receiver tests. The distortion of the audio signal, measured as SINAD (signal to noise and distortion), is a measure of the receive quality at the respective power level. The higher the value, the better the receive quality.

The level of the audio signal is an indicator of the presence of the signal.

How to read the 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.

RX SINAD

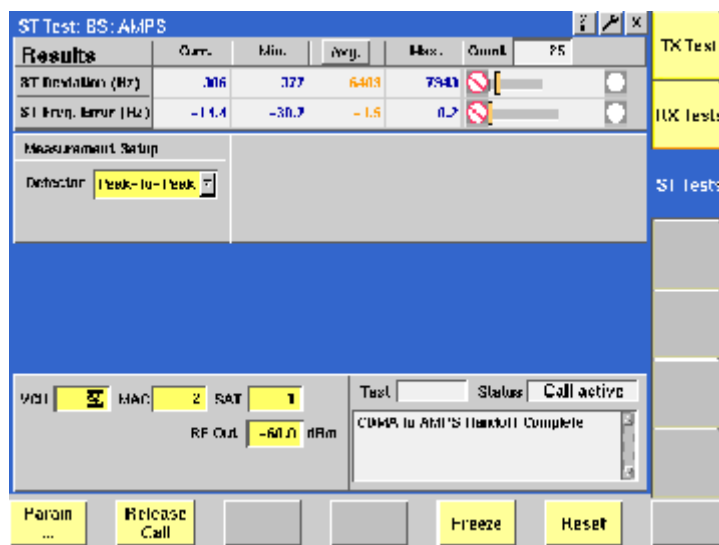
The SINAD, measured in dB, is a measure of the receive quality. The higher the value at low power levels, the better the quality or receiver sensitivity. The measurement is taken from the signal fed into the **AF IN** connector.

AC Level

The AC level indicates if the wanted audio signal for which SINAD is measured is still present. The 4400 measures the root mean square-valued voltage of the signal at the **AF IN** connector.

ST Tests

This test menu is designed to provide you with the confidence that the signaling functions of the mobile phone are operational. During the AMPS call, press the [ST Tests] marker tab to enter this menu.



What it does The signaling tone (ST) is a 10 kHz tone used in AMPS for mobile ringing, call terminations, handoffs, and switch-hook operation. The ST test is designed to verify and troubleshoot the baseband signaling performance and circuitry of a mobile.

The 4400 sends an alert message to the phone, which causes the mobile to enable the ST tone in addition to ringing. Do not answer the call immediately because the measurement results in the ST test are valid only while the phone is being paged.

How to read the 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 relative to the measurement limits.

| | |
|-----------------------|---|
| ST Deviation | This row indicates the deviation of the signaling tone. The typical deviation is 8 kHz \pm 0.8 kHz for positive and negative peak measurements (see below). |
| ST Freq. Error | The frequency error of the signaling tone should be close to zero. |

Selecting measurement-specific parameters **Measurement Setup: Detector** — This setting affects how frequency deviation is measured. The 4400 can either measure the positive peak (Peak+), the negative peak (Peak-) or the difference between the positive and negative peak (Peak-to-Peak).

Parameters

Setting up the 4400 for AMPS tests is easy once the correct CDMA parameters are selected. Results are affected by a few parameters as described above with the test menus.

For a quick overview if a mobile phone is within specified tolerances, the 4400 provides pass/fail indications. The limit values are accessible by a push on the **Param...** softkey while on an AMPS channel.

The Limits softkey provides access to a menu for the various test limits. 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.

NOTE

The Limit menu includes an **Undo** function. By pushing this softkey, all parameter values will return to their respective state before calling up the menu.

NOTE

The Limit menu includes a **Default** softkey, setting the parameters on the screen to their default (factory) settings.

The 4400 can display 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.

While on an AMPS channel, push the **Param...** softkey. The Limits menu will appear with the following sections:

- RX
- TX Deviation
- TX Freq. Error
- TX Power

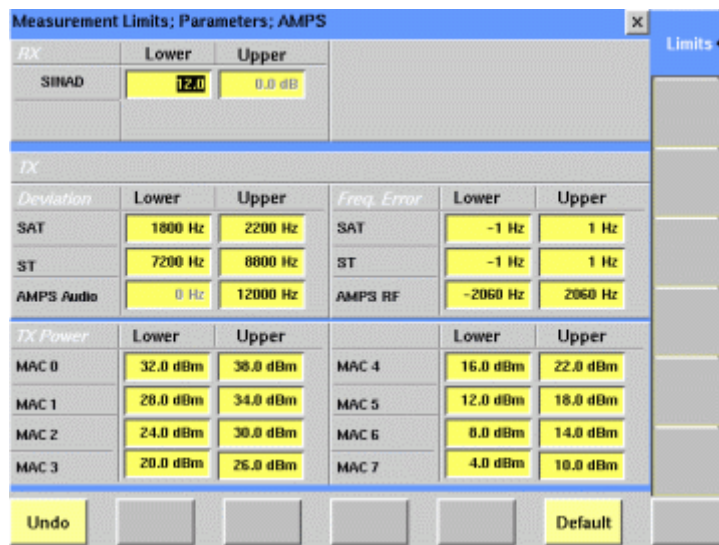
How to read and enter the measurement limits

For most measurement parameters, 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.

If a measurement can have positive result values only, the lower limit is zero in most cases. For these measurements, only an upper limit (maximum) can be entered.

Limits



| | |
|-----------------------------|--|
| SINAD | The upper limit does not make any sense because the higher the ratio between signal on the one hand and noise and distortion on the other, the better. The lower limit depends on the measurement conditions, in particular on the RF input level at the mobile phone. |
| SAT Deviation | The nominal SAT deviation is 2 kHz with a tolerance of ± 0.2 kHz, so the lower limit is 1800 Hz and the upper limit is 2200 Hz in most cases. |
| ST Deviation | The typical deviation is 8 kHz ± 0.8 kHz for positive and negative peak measurements, so the lower limit is 7200 Hz and the upper limit is 8800 Hz in most cases. |
| AMPS Audio Deviation | The upper and lower limits of the audio deviation depend on the signal fed into the mobile phone. The upper limit should not exceed 12 kHz as this is the maximum allowable value for AMPS phones. |
| SAT Freq. Error | The default SAT frequency error tolerance is ± 1 Hz. |
| ST Freq. Error | The default ST frequency error tolerance is ± 1 Hz. |
| AMPS RF Freq. Error | This is the input field for the frequency error limit. The frequency error is displayed on the TX Test menu. The default tolerance is ± 2060 Hz. |
| MAC 0 thru 7 | The nominal MAC power levels are listed in Table 15 on page 117 . The default lower limit is set to 4 dB below the nominal value and the upper limit is set to 2 dB above the nominal value. Note that the MAC 0 and MAC 1 limits should be changed if the mobile only supports a lower power class! |

AMPS Nonsignaling Mode

5

This chapter provides task-based instructions for using the nonsignaling mode (Generator/Analyzer) of AMPS within the 4464 CDMA2000 System Option. Topics discussed in this chapter are as follows:

- "Overview" on page 130
- "RF generator" on page 130
- "Transmitter tests" on page 132
- "Receiver tests" on page 133
- "Signaling tone tests" on page 134
- "Common elements of the RF analyzer menus" on page 135
- "Parameter menus" on page 136

Overview

The generator/analyzer mode of the 4400 provides test and measurement tools for measurements while the mobile phone is in asynchronous operating or test mode; these are measurements without requiring a call setup. Both RF generator and analyzer can be operated independently.

To gain access to the menus of the AMPS generator/analyzer tool:

- 1 Go to the Welcome menu (see picture below).
- 2 Push the marker tab for the corresponding system option (AMPS).
- 3 Push the **Gen/Ana** softkey.



This section consists of the following parts:

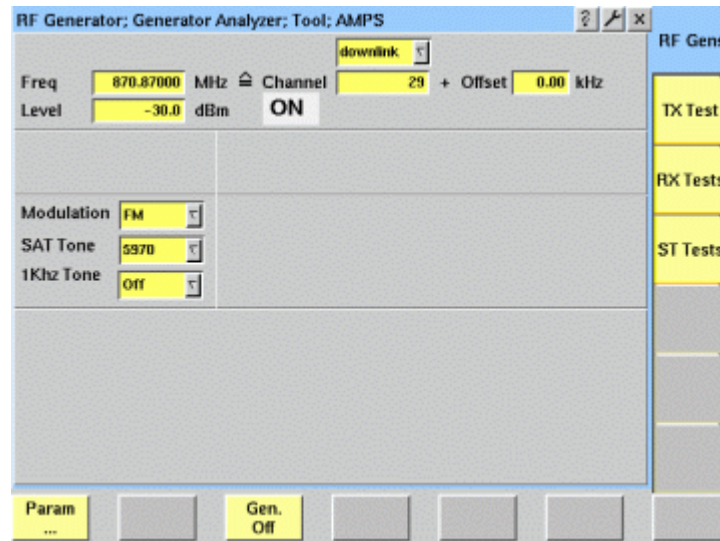
- [RF generator](#) – This section describes the menu of the RF signal generator.
- [Transmitter tests](#) – This menu provides measurement results for the modulation quality.
- [Receiver tests](#) – The receiver quality of the mobile phone is measured and displayed.
- [Signaling tone tests](#) – This menu shows the quality of the signaling tone transmitted by the mobile phone.

RF generator

To gain access to this menu, push the [RF Gen] marker tab while working with the generator/analyzer.

This menu allows you to set the parameters of the RF generator.

The RF generator can be set to generate radio signals at a defined carrier frequency and modulation. AMPS uses frequency modulation, so the signal can be either FM or totally unmodulated in order to stimulate the RF circuitry.



What it does

The 4400 can be set up to generate a typical voice carrier as from a base station. The generator can be used to test the receiver of the mobile phone in a special test (engineering) mode. It can also be used to provide a frequency reference which the phone may require before transmitting any signal in test mode.

Parameters

Freq — Entry field to set the output frequency of the RF generator. Entry ranges: 430 to 500 MHz, 800 to 1000 MHz and 1700 to 2300 MHz in increments of 10 Hz.

uplink/downlink — Selection field to specify either the uplink (reverse link) or the downlink (forward link) frequency range. Selections possible: `uplink` or `downlink`.

Channel — Entry field for the AMPS channel number. If you enter a new value in this field, the Frequency field will be recalculated based on the uplink/downlink input fields provided that the channel number exists. Entry range: 1 to 1023

Offset — Entry field for the offset from the channel frequency specified. Entry range: -99.99 to 99.99 kHz in increments of 10 Hz.

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 0.0 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 Ω.

ON/OFF — Display field for the current output status of the RF generator. When ON is displayed, RF signals are output; when OFF is displayed, the RF generator is silent.

To toggle between the two states, use the **Gen. Off** or **Gen. On** softkey, respectively.

Modulation — Selection field to turn the modulation on or off.

Selections possible: FM or none. If set to FM, the 4400 will transmit a carrier with SAT (supervisory audio tone) and (if selected) a 1 kHz tone.

NOTE

Setting “none” means that the RF output signal generated by the 4400 will be unmodulated, i.e. a pure carrier sine wave.

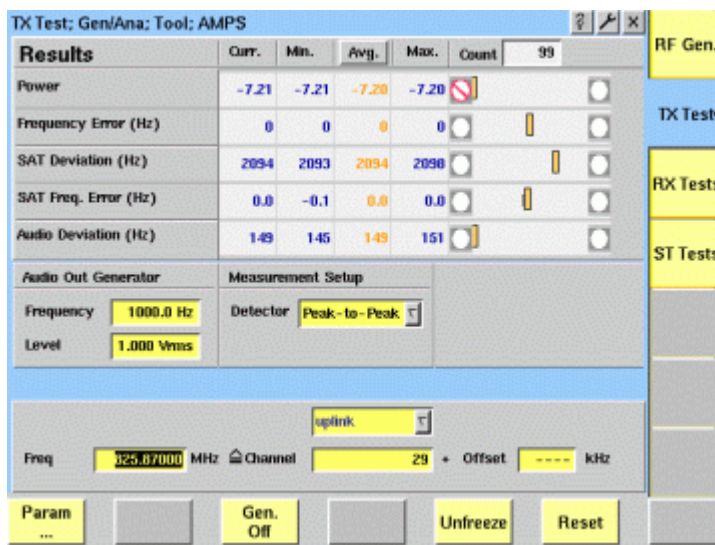
Transmitter tests

To gain access to this menu, push the [TX Test] marker tab while working with the generator/analyzer.

This menu displays the results of the modulation 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 an AMPS carrier with SAT and possibly an audio tone. The expected SAT depends on the SAT setting for the RF generator.



The RF analyzer takes measurements in asynchronous mode, without any call processing functions.

If a [Coupling loss](#) table is defined and activated in the appropriate CDMA menu, it 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 this parameter in the lower area of the menu, just above the softkeys.

The measurement results are similar to the corresponding areas and fields of the AMPS calls & measurements test menu described in section ["TX Tests menu" on page 121](#).

The frequency parameters and the softkeys are explained in section ["Common elements of the RF analyzer menus" on page 135](#).

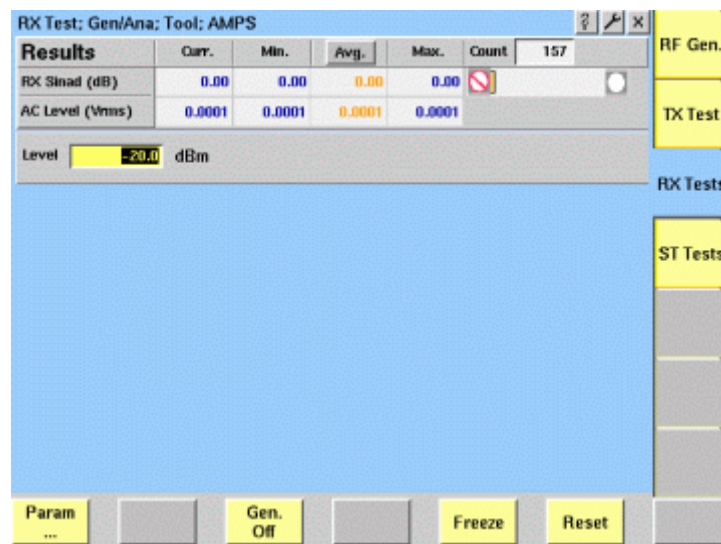
Receiver tests

To gain access to this menu, push the [RX Tests] marker tab while working in generator/analyzer mode.

This menu displays the results of the receiver measurements of the RF signal received at the RF in/out connector while in nonsignaling mode.

NOTE

The receiver measurements are only supported with the 4470 Audio Option installed.



The receiver measurements assume that the mobile phone under test is in asynchronous mode, without any call processing functions active.

If a [Coupling loss](#) table has been activated in the CDMA menus, it will be used for AMPS measurements as well.

The measurement results are similar to the corresponding areas and fields of the AMPS calls & measurements test menu described in section ["RX Tests menu" on page 123](#).

The softkeys are explained in section ["Common elements of the RF analyzer menus" on page 135](#).

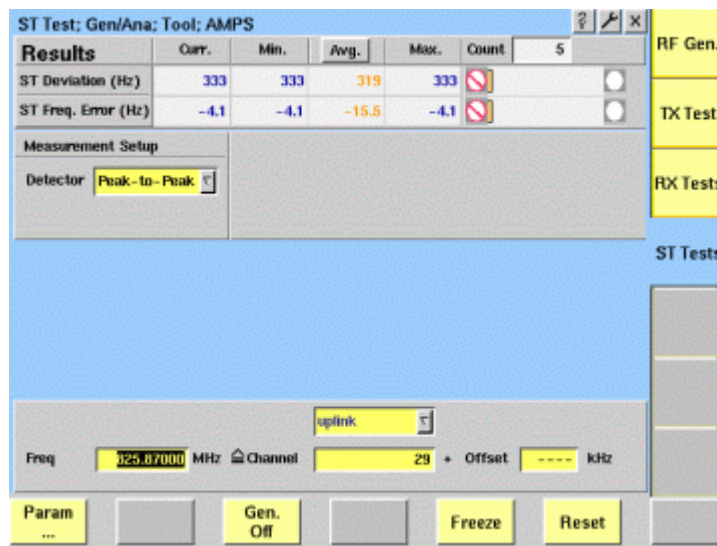
Signaling tone tests

From any of the AMPS generator/analyzer menus, push [ST Tests] to enter the menu.

This menu displays the results of the signaling test measurements of the RF signal received at the RF in/out connector while in nonsignaling mode.

NOTE

The analyzer will measure the signal and display the results assuming that the signal is an AMPS carrier with signaling tone and possibly an audio tone.



The signaling tone measurements assume that the mobile phone under test is in asynchronous mode, without any call processing functions active.

If a [Coupling loss](#) table has been activated in the CDMA menus, it will be used for AMPS 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 ST Test menu.

The measurement results are similar to the corresponding areas and fields of the AMPS calls & measurements test menu described in section ["ST Tests" on page 124](#).

The frequency parameters and the softkeys are explained in section ["Common elements of the RF analyzer menus" on page 135](#).

Common elements of the RF analyzer menus

Entry fields of the RF analyzer

The radio frequency to measure on is set in this menu area.



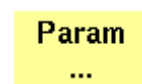
Freq — Entry field to set the measurement frequency for the RF analyzer.
Entry range: 430 to 450 MHz, 800 to 1000 MHz and 1700 to 2300 MHz in increments of 10 Hz.

Channel — Entry field for the AMPS channel number.
If you enter a new value in this field, the Frequency field will be recalculated based on the uplink/downlink input fields provided that the channel number exists.
Entry range: 1 to 1023.

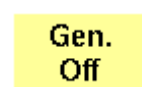
Offset — Entry field for the offset from the channel frequency specified.
Entry range: -99.99 to 99.99 kHz in increments of 10 Hz.

Avg — Click on this field or push the **Enter** key on this field to open a dialog box. The box allows you to define the number of last measurement results over which to average.
Select **OK** to confirm the current value or **Cancel** to return to the setting that was in use before opening this dialog box.

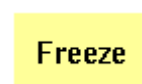
Softkeys of the RF analyzer



This softkey leads you to the parameter screens where you can set up the limits.



A push on this softkey switches the [RF generator](#) either on or off.



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.



A push on this softkey resets the statistic evaluation of the measurement results. The measurement counter will be reset as well.

Parameter menus

The Parameter menus allow you to access additional parameters relevant in both nonsignaling and in call mode at a more detailed level, which need not be changed in everyday use. To access the Parameter menus, press the **Param...** softkey on any of the generator/analyzer menus and then press the appropriate marker tab.

More details about the parameter menus can be found in section "[System parameters](#)" on page 73.


Tools

6

This chapter provides task-based instructions for additional tools within the 4464 CDMA2000 System Option. Topics discussed in this chapter are as follows:

- ["Overview" on page 138](#)
- ["Configuration" on page 138](#)
- ["Utilities" on page 151](#)
- ["Audio measurements" on page 166](#)

Overview

This chapter shows you the functionality in the Tools section of the user interface. 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 138](#) lets you view and change the instrument configuration. This includes software updates and upgrades and interface settings.
- Section ["Utilities" on page 151](#) includes general functions such as a configurable power supply and current measurements.
- ["Audio measurements" on page 166](#) shows you how to operate the audio signal generator and analyzer.
-

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.

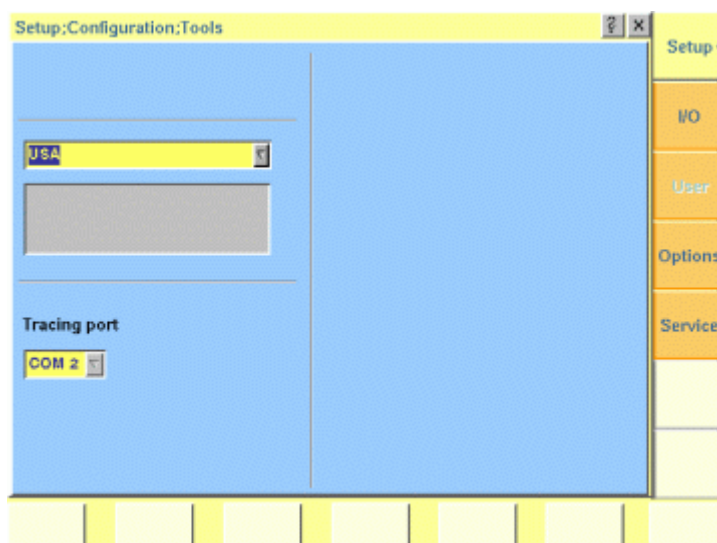
This section consists of the following parts:

- ["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

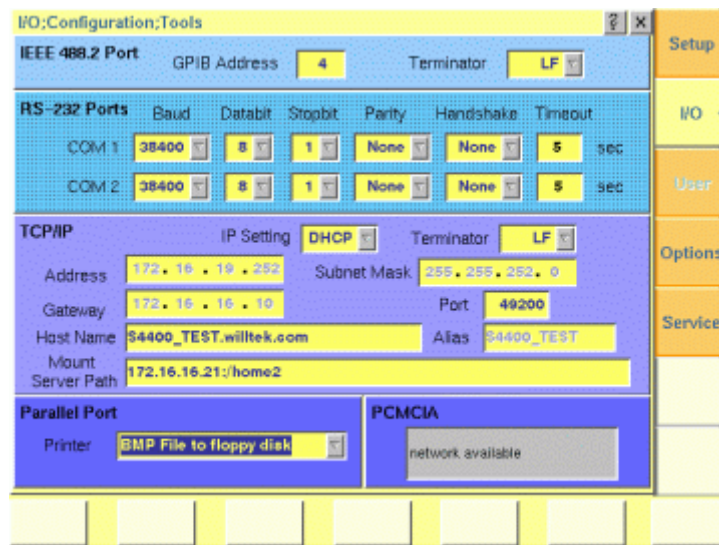
NOTE

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 148)
- 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 | Terminator |
|-------------------|-------------------------|-------------|---------------|
| 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 computer, that computer must run an NFS client software. Most computers running 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 accessible 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:

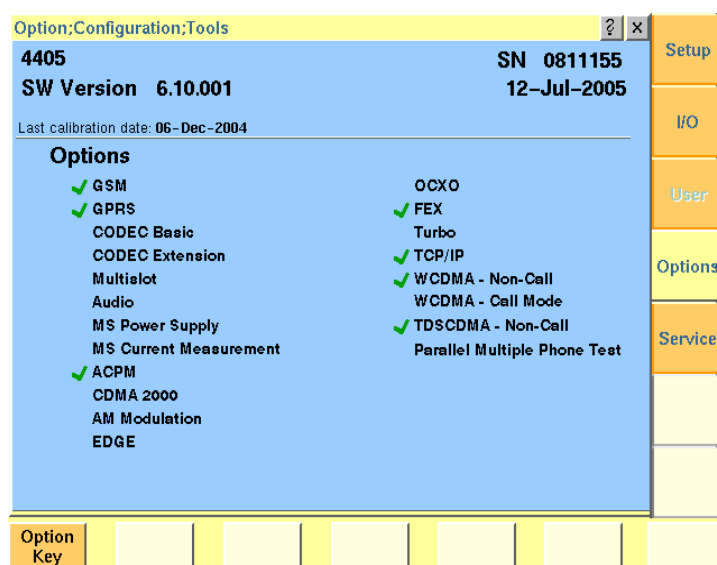
- HP DeskJet 400 (default)
- HP DeskJet 680
- HP Laserjet
- Canon
- Epson Stylus Color
- Epson Stylus Color II
- Epson Stylus Color IIs
- Epson Stylus Color 200
- Epson Stylus Color 400
- Epson Stylus Color 500
- Epson Stylus Color 600
- Epson Stylus Color 800
- Epson Stylus Color 1500
- Epson Stylus 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.

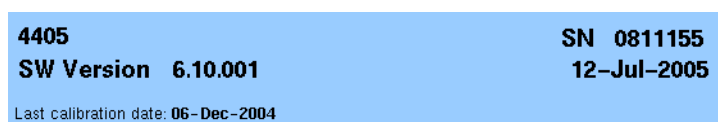


This menu consists of two areas:

- 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

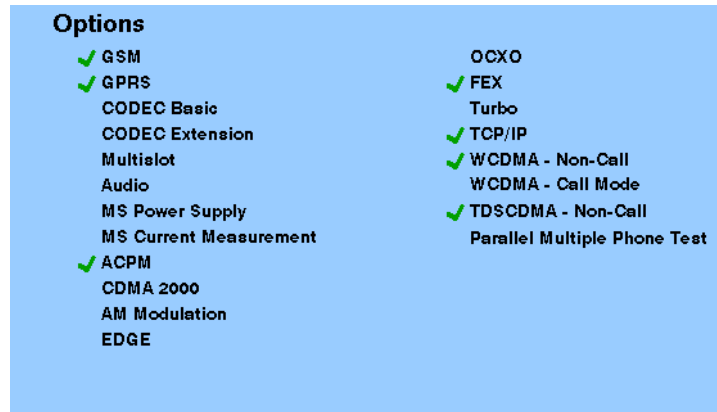
The availability of an option may be dependent on the serial number.

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.



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.

Parallel 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.

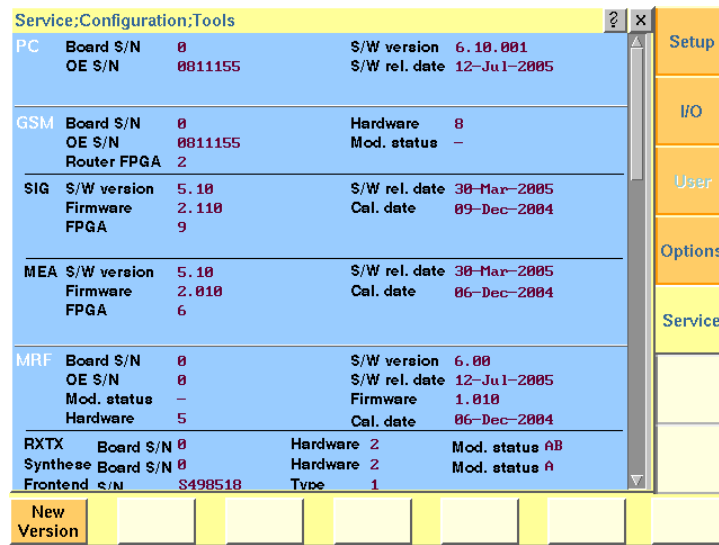
Options that are software-based only (like ACPM for instance) may be installed on-site. The only thing required then is a special code available from Willtek. When you have the code at hands, follow the instructions below:

Option Key

- 1 Push this softkey.
- 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

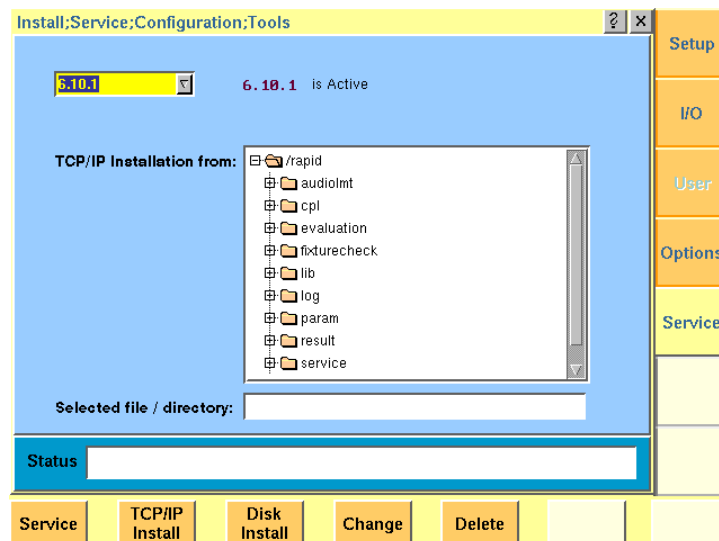
Willtek seeks to permanently improve its products and makes new software versions available on the Internet at www.willtek.com. Several different methods exist to upgrade the software of your 4400: From floppy disks, from a remote PC over the LAN (TCP/IP required) and from a remote PC over GPIB.



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.

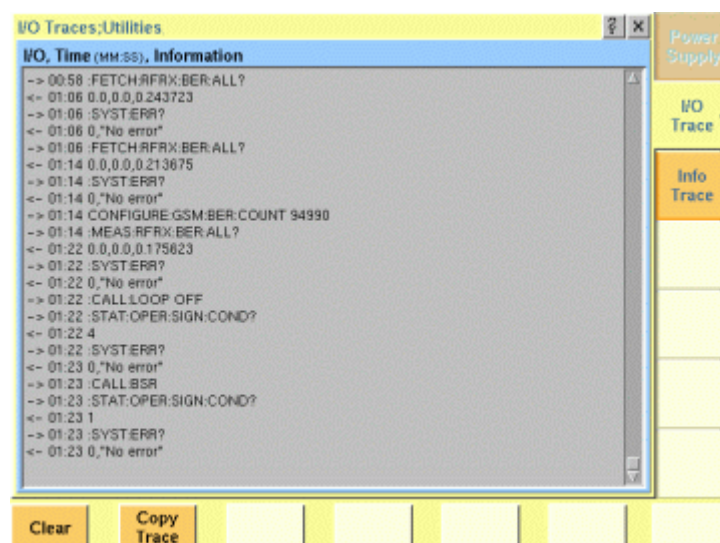
Access to the Utilities menus

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

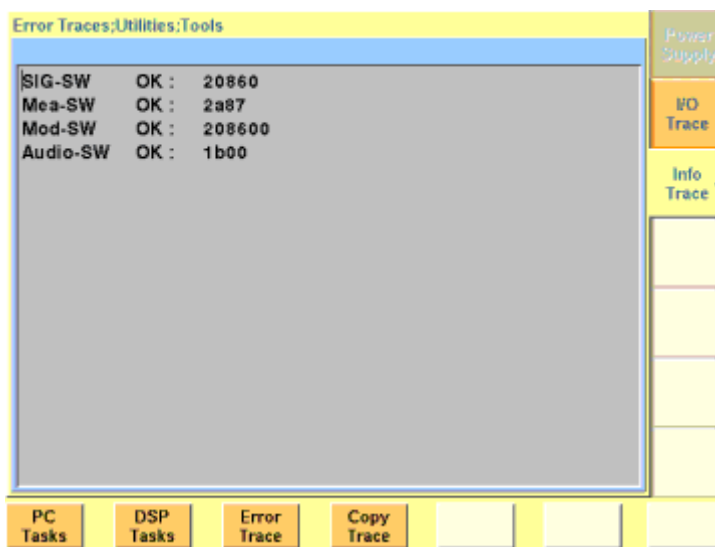


Resets the entire I/O Trace menu (all entries in all areas of the menu will be cleared).



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.

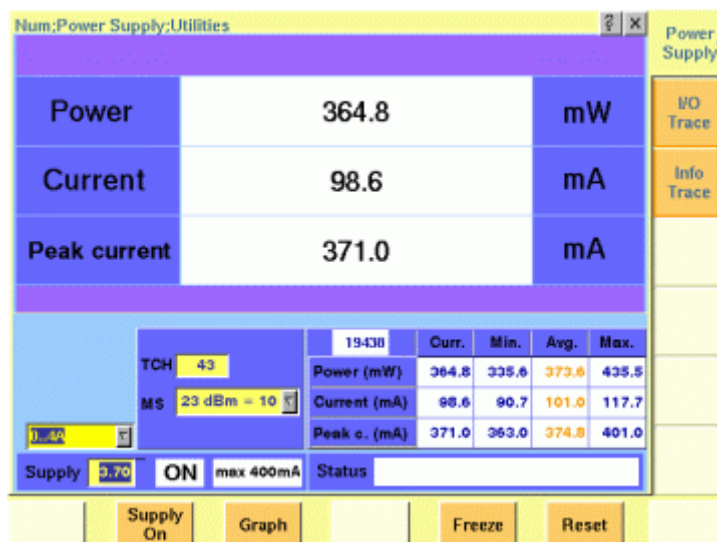
MS power supply

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 Supply 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.

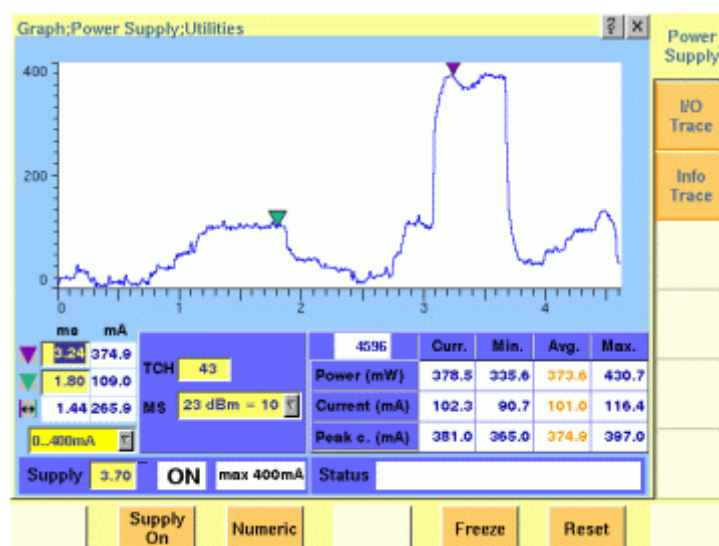
In combined graphical/numerical mode, the menu shows a graph with the current versus time. The scale of the vertical axis is mA and the horizontal axis scaling is in ms (milliseconds).

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 $(3+3) * 0.54738 \text{ ms} = 3.28 \text{ ms}$.

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 177.

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 spectrum analyzer menus provide test capabilities to evaluate the RF signals received by the 4400 in the frequency domain.

Two different measurement menus are available: Modulation spectrum and ACPM due to modulation.

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 ACP measurements of the spectrum due to modulation and wideband noise.
- “[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.

Access to the spectrum measurement menus

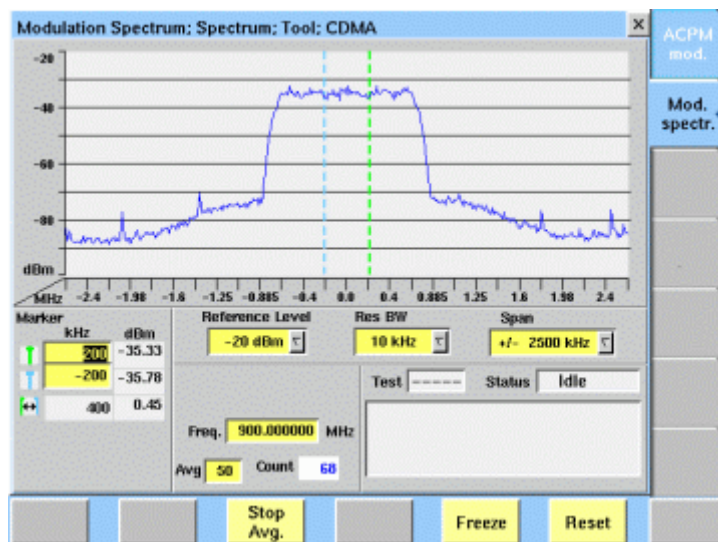
Spectrum measurements are possible in all relevant operating states of the 4400.

- 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 **CDMA traffic channel**, select marker tab [CDMA] 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.

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 parameter 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 scaled in dBm, the horizontal axis in MHz.

The selection field *Reference Level* allows you to set the maximum power level to optimize the vertical resolution.

Selection fields *Res BW* and *Span* allow to set the resolution bandwidth and frequency span for the horizontal axis of the live display.

Reference 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 on the span of the live display. The following combinations are available:

- For the ± 500 kHz span, resolution bandwidths of 2, 5 and 10 kHz are available.
- For the ± 1250 kHz span, resolution bandwidths of 5, 10 and 30 kHz are available.
- For the ± 2500 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: ± 500 kHz, ± 1250 kHz and ± 2500 kHz.

Setting the center frequency

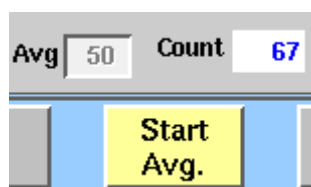
The center frequency of the spectrum analyzer is the frequency of the expected uplink carrier.

- In synchronous mode (i.e. when a traffic channel has been set up), this frequency is given by the traffic channel number used. The corresponding frequency is displayed on-screen in the Freq. 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 can be changed using the Freq. input field.

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

Start 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.

Important to note

- The entry range for the Avg entry field is from 1 to 100.
- 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 modulation spectrum measurements

The Adjacent Channel Power Measurements for the 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 CDMA specifications.

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,
- baseband filter, and
- I/Q modulator.

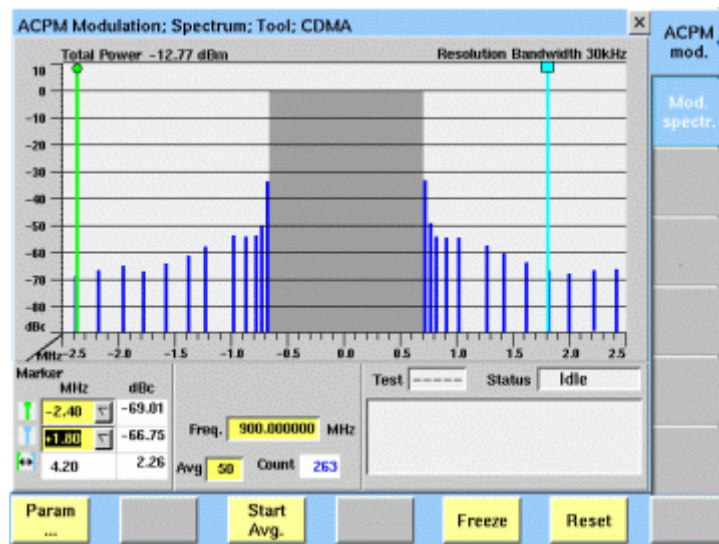
The following defects of the mobile might be detected with this measurement:

- neighboring 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 while in the spectrum menus.



On first sight

The mobile's modulation spectrum is displayed graphically in the menu's live display for the discrete frequencies specified by the CDMA specification. 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, and the total power measured appears at the top of the live display.

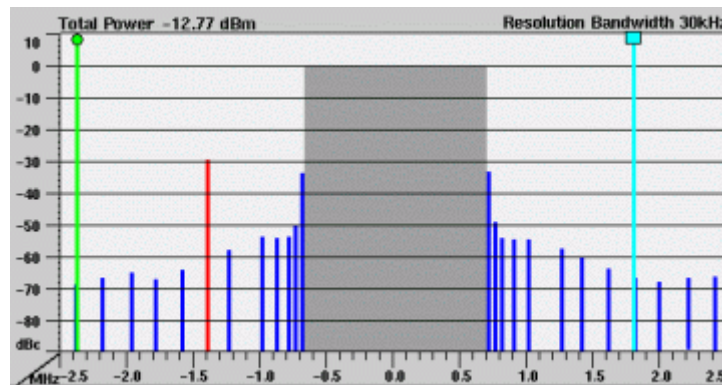
Understanding the measurement principle

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 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 ± 700 kHz, ± 750 kHz, ± 800 kHz, ± 890 kHz, ± 1000 kHz, ± 1250 kHz, ± 1400 kHz, ± 1600 kHz, ± 1800 kHz, ± 1980 kHz, ± 2200 kHz, and ± 2400 kHz offset 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. All levels displayed in the live display are relative to the reference level (at the carrier frequency) and thus given in dBc.

How to read the live display

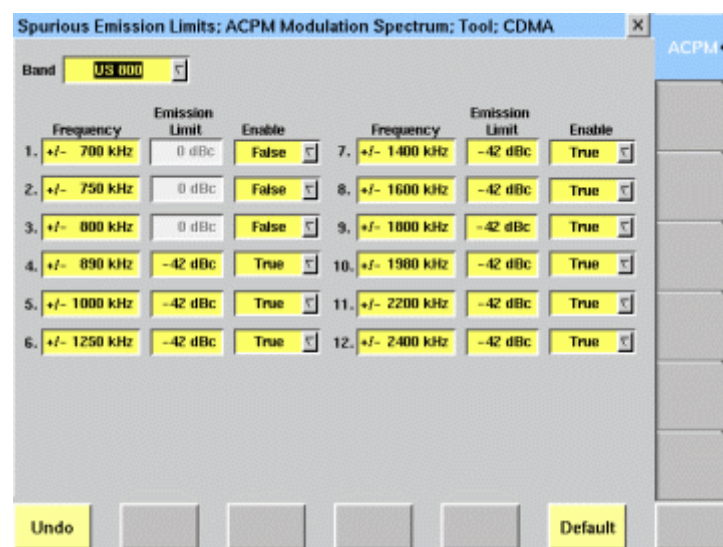
User-definable measurement frequencies and test limits for each frequency can be defined in the Parameters menu. Measurement points can be set in the range from 700 to 2400 kHz; the gray area marks the in-channel band in which adjacent channel power measurements cannot be taken.

When the measurement result for an individual frequency is within the limits, the corresponding vertical bar will be displayed in blue. A vertical bar displayed in red indicates that the current measurement result is beyond the limits.

Setting measurement frequencies and limits

The measurement frequencies and corresponding limit values can be defined in the Parameters menu for the ACPM menu.

- 1 In the ACPM Mod. menu, press the `|Param|` softkey.
A new menu appears.



- 2 Enter the frequencies which you want to measure. Valid entries are in the range from 700 to 2400 kHz with 10 kHz resolution. Up to twelve frequencies can be defined.
- 3 If you want to define an upper limit for the measurement frequency, move the cursor to the corresponding Enable field in the row and set the Enable parameter to True.
- 4 If appropriate, enter an upper limit in the Emission Limit column after enabling limit checking. The limits are expressed in dBc, relative to the power at the center frequency.

Setting the center frequency

The center frequency of the spectrum analyzer is the frequency of the uplink carrier.

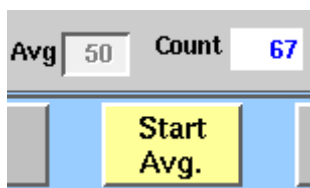
The center frequency cannot be modified in the RF spectrum analyzer menus.

- In synchronous mode (i.e. when a traffic channel has been set up), this frequency is given by the traffic channel number; the corresponding frequency is displayed in the Freq. display field.
- In asynchronous mode (i.e. when there is no signaling exchange between the mobile under test and the 4400), the center frequency can be entered directly in the Freq. entry field.

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

- Start 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 100.
- 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.

Softkeys The softkeys of the RF spectrum measurement menus provide access to special functions and features.

Start Avg.

A push on this softkey starts the averaging tool of the spectrum analyzer.

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. |.

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.

Display markers

| 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.

Audio measurements

The Audio Option provides an AF generator and an AF analyzer.

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.

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 (software switch). The input impedance is approx. $250\text{ k}\Omega \parallel 20\text{ pF}$. The usable AC frequency range stretches from 30 Hz to 20 kHz.



HAZARD

The absolute maximum input voltage is $\pm 40\text{ V DC}$ or 30 V (rms) AC . An AF input voltage of more than $\pm 40\text{ V DC}$ 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 unbalanced.
The input impedance is $< 1 \text{ k}\Omega$.
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 unbalanced.
Best performance is achieved on a 600Ω 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 8Ω ; 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.

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.

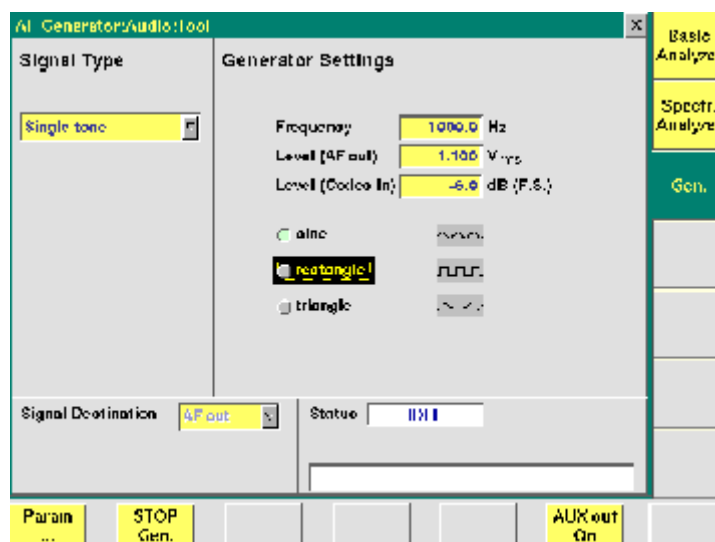
The generated signal can be fed to the analog outputs ("AF out" or "AUX 4 (AUX out)").

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.

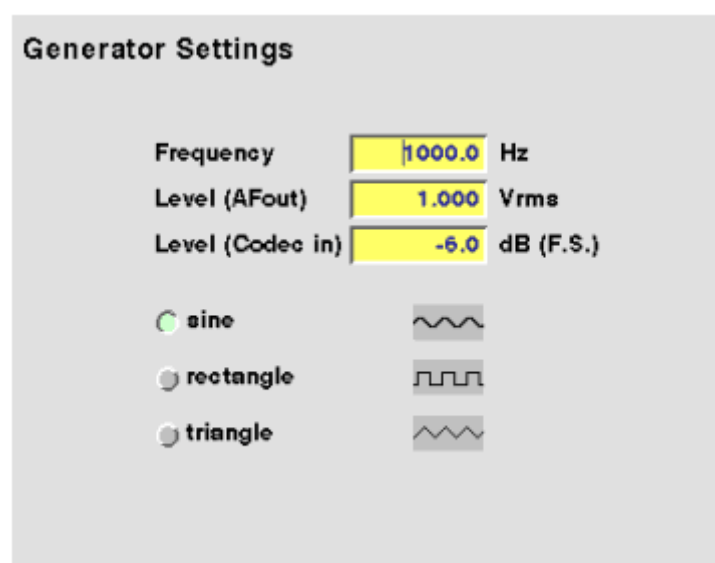


This menu consists of four areas:

- 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.

sine — Check box. If ticked, the output signal will have a sinusoidal waveform.

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.

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. 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

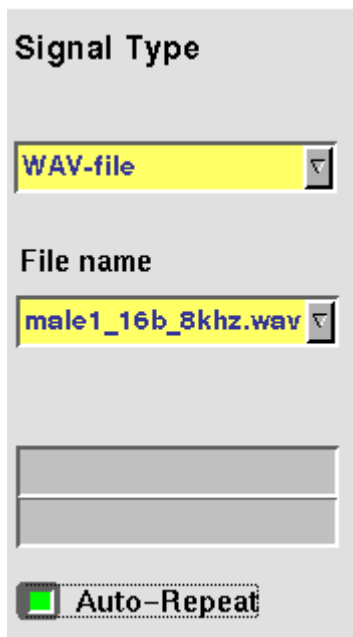
Wave files can hold various sampling rates and data widths. The 4400 supports the following formats:

- 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.

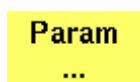
NOTE

This field is visible only when wave file replay has been selected as the signal type.

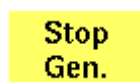
Auto-Repeat — Check box. If ticked, the signal from the file is repeated over and over again until the generator is stopped with the **STOP Gen.** softkey.

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.

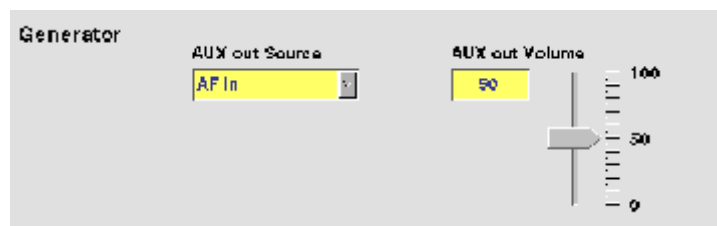
Aux out On

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.

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

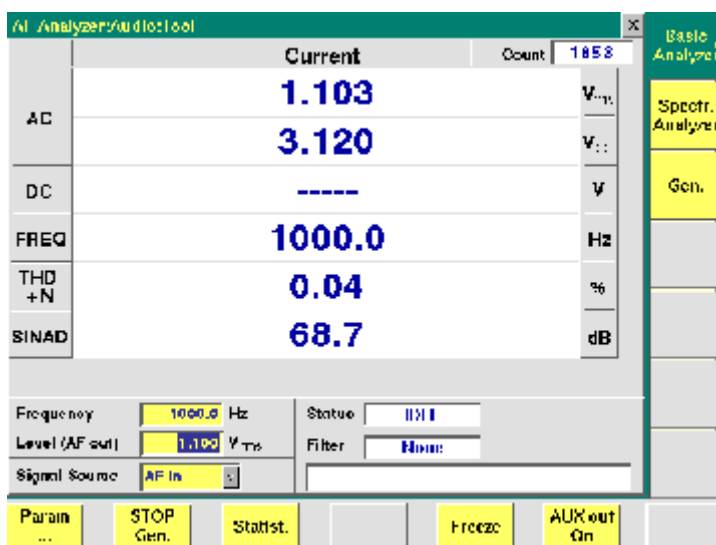
The 4400 provides two different audio analyzers:

- 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, in V_{pp} (volts, measured peak-to-peak) and V_{rms} (volts, RMS-valued) .

DC — Display field. If DC coupling is selected, this display field shows the DC voltage component of the analog audio signal measured.

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 Signal to Noise And Distortion.

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.

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.

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 is 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.

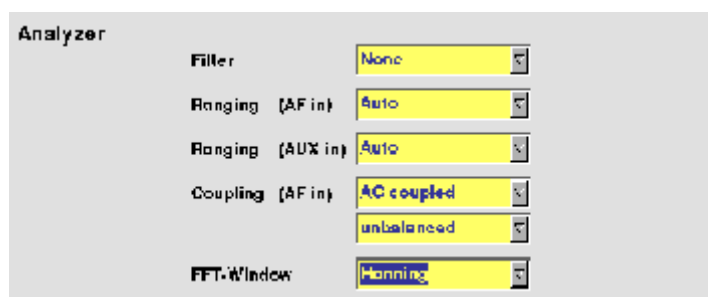
Aux out On

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`.

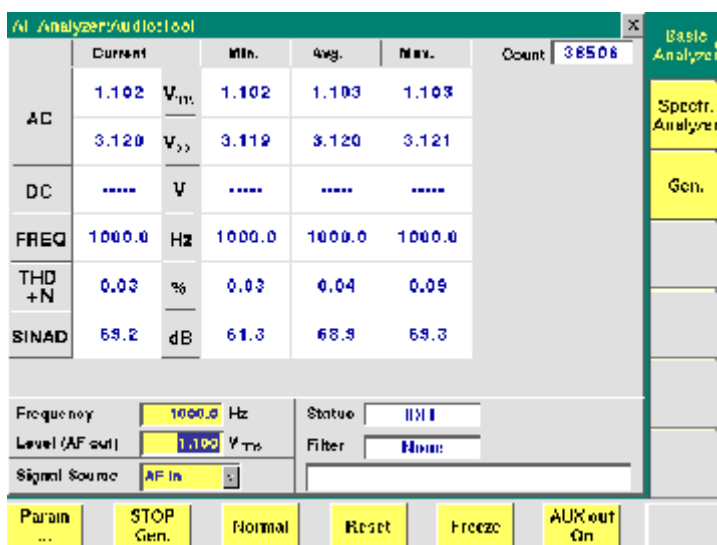
- `unbalanced` means there is one 'hot' signal line and one ground line. This is sometimes also referred to as 'asymmetric' or 'single ended'.
- `balanced` means that there are two 'hot' signal lines and no ground line (there may be a shield). This is sometimes also referred to as 'symmetric', 'differential', 'bridged' or 'floating'.

FFT-Window – Selection field. Allows to select the window for the Fast Fourier Transformation 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.

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 Signal to Noise And Distortion.

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.

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.

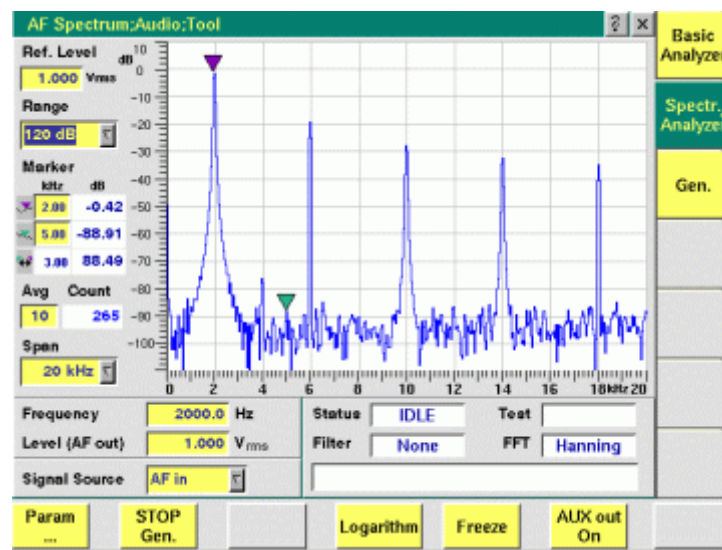
AF spectrum analyzer

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,
- .

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).

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.

| Marker | kHz | dB |
|--------|------|---------|
| 1 | 1.00 | -89.57 |
| 2 | 5.00 | -104.19 |
| | 4.00 | 14.62 |

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.

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.

RAPID!

7

This chapter describes how to use and how to program RAPID! applications. Main topics described in this chapter are as follows:

- "Overview" on page 184
- "Using RAPID!" on page 184
- "RAPID! syntax" on page 201
- "Commands" on page 209
- "Functions" on page 234
- "Tables" on page 252

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 (Beginner's All-purpose Symbolic Instruction Code) 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

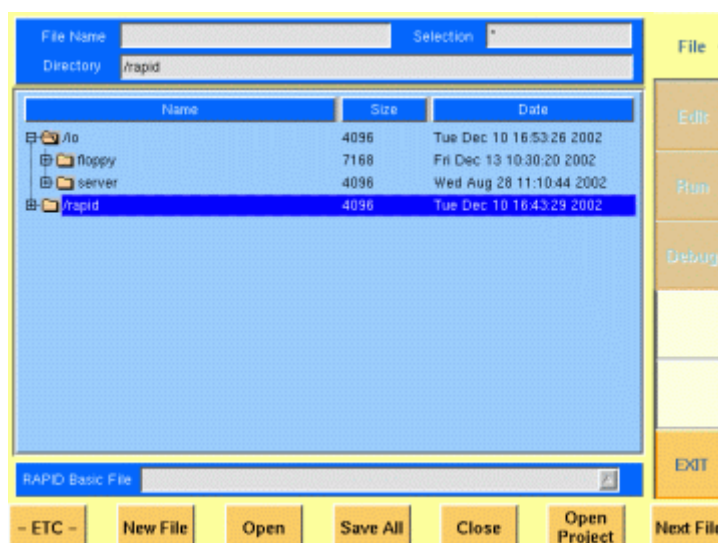
SCPI commands are used for configuring the 4400 and for executing and evaluating measurements. You'll find more about the use of SCPI commands in subsection "[SCPI and RAPID!](#)" on page 269.

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.

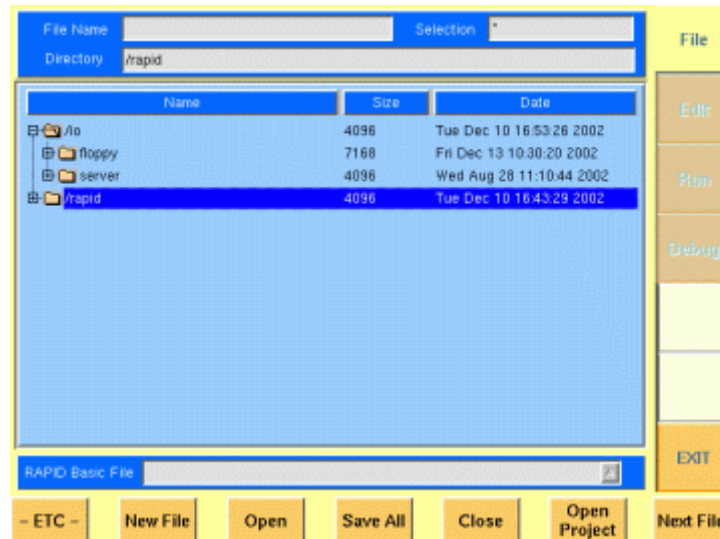


Quitting the RAPID! environment.

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

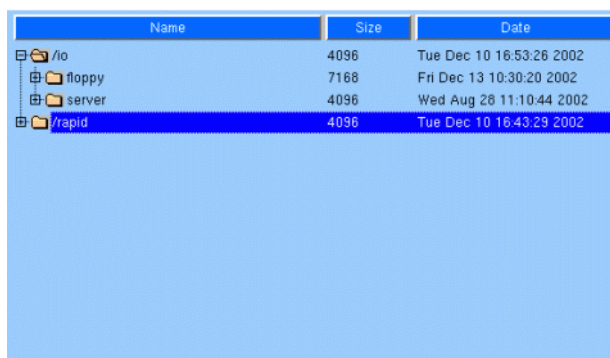
The directory path displayed here is the same as selected in the browser area, as long as you do not change it.

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:

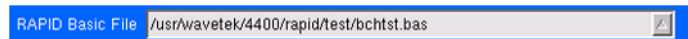
- **/io**. This directory includes the **floppy** subdirectory which in turn carries the contents of the floppy disk (if any is inserted in the floppy drive). An additional subdirectory, **server** may be available if the 4400 is equipped with an Ethernet PC card and the TCP/IP Option and if a server 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.



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.



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.

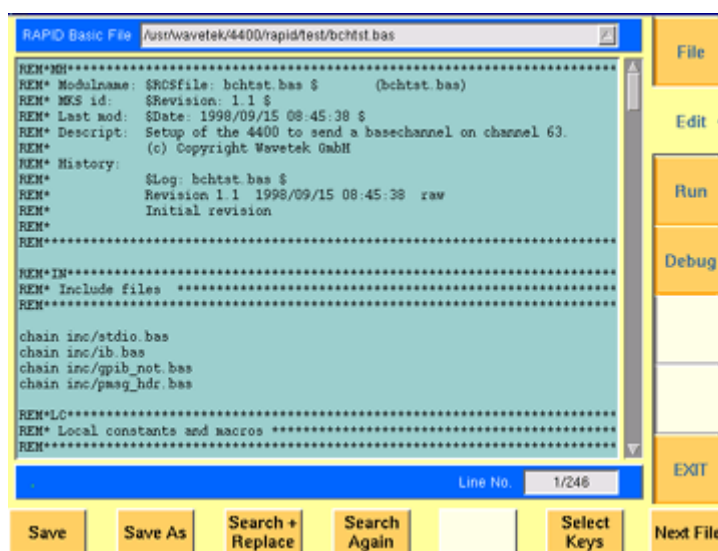
NOTE

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



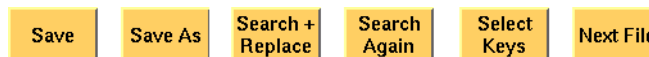
In the status area, the 4400 shows the following information:

- **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 — 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](#).

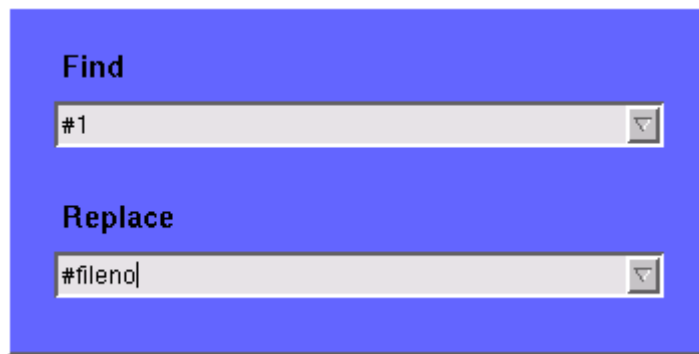
Search Again — Continues the search of the text passage as entered in the [Search/Replace menu](#) before. The direction of the search is "downwards", i.e. from the current cursor position towards the end of the file.

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 (**F**ind).

If the text found shall be replaced by another text, enter the new text in the lower line of this menu (**R**eplace).

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 **F**ind 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 **F**ind selection field and replaces it with the text entered on the **R**eplace selection field. The direction of the search is "downwards", i.e. from the current cursor position towards the end of the file.

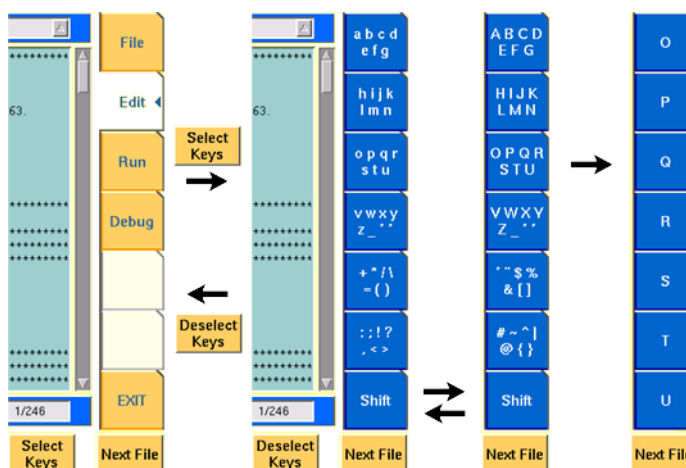
Replace All — Searches for all occurrences of the text entered on the **F**ind selection field and replaces every single one with the text entered on the **R**eplace selection field. The direction of the search is "downwards", i.e. from the current cursor position towards the end of the file.

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 'type-writer' 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

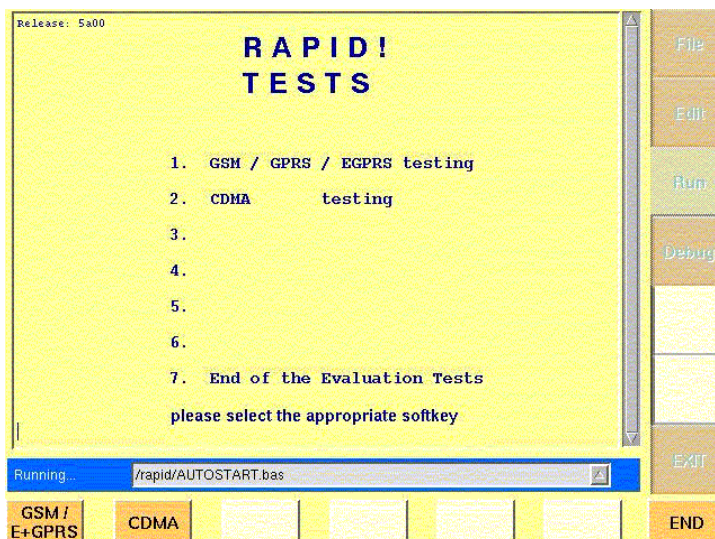
Softkey **Deselect Keys** takes you back to the usual marker tabs of the RAPID! environment.

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.

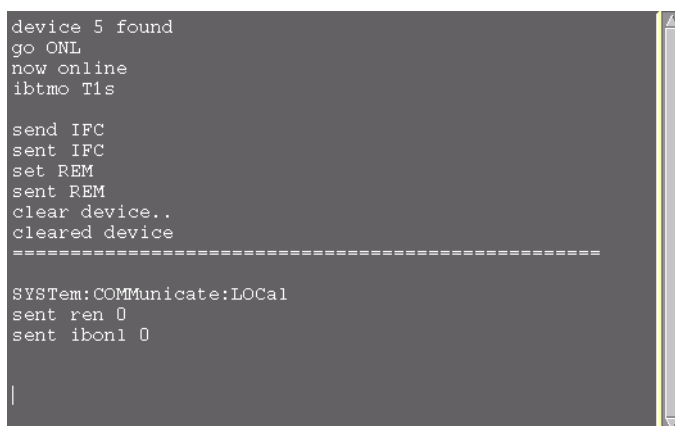
The softkeys provide access to important functions of the Run menu, like starting a RAPID! program.

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:

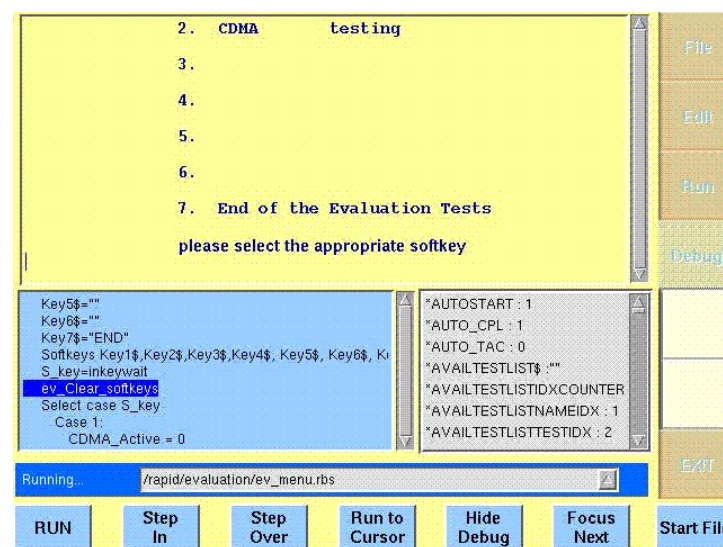


RUN — The RAPID! file, currently selected on the RAPID Basic File selection field, will be loaded and the program execution started.

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_PARAM

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 : 96
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.



- 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.

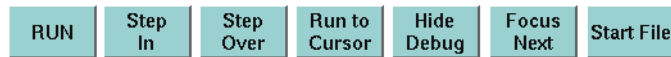


- 4 Enter the new value for the selected variable in the `edit watch:` dialogue box, using the 4400's numeric keys or the external keyboard.
- 5 Confirm the value entered by pushing the **Enter** function key on the 4400 or by pushing the external keyboard's **Enter** key.
- 6 The new value will now 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.



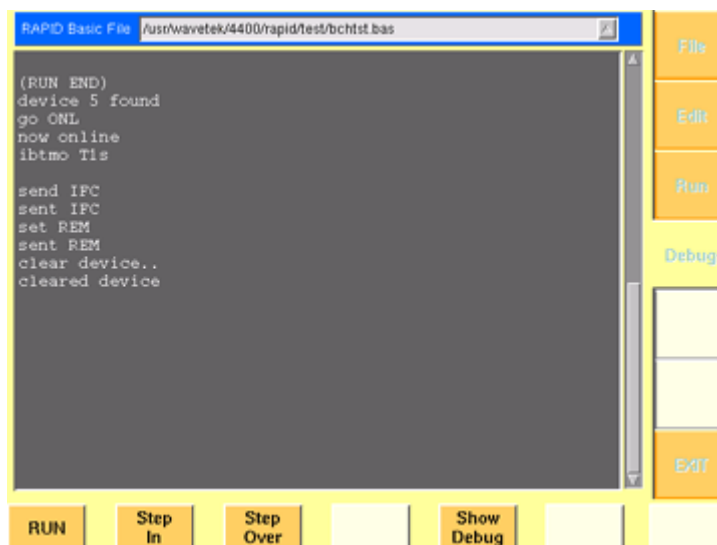
RUN — Loads the program currently selected on the RAPID Basic File selection field and starts execution of the program.
In this mode, the 4400 will highlight the last command performed in the Source area.

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 derivatives, 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 lowercase 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 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, μ W, mW, W |
| Angles | DEG or $^{\circ}$, 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 | <code>{0...9} [.{0..9}]</code> | 123.456 | -1.79769313486232E308 to -4.94065645841247E-324 for negative values and from 4.94065645841247E-324 to 1.79769313486232E308 for positive values |
| Binary | <code>&tB {0...1}</code> | <code>&tB100111011</code> | 2 to the power of 32 (2^{32}) |

| | | | |
|-------------|---------------------|----------|---|
| Octal | &tO {0...7} | &tO37721 | 2 to the power of 32 (2 ³²) |
| Hexadecimal | &tH {0...9 A...F} | &tH4AF3 | 2 to the power of 32 (2 ³²) |

Names of constants without any identifier (&tB, &tO, &tH) 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, then 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 operators | = | 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 <i>stringExp</i> |
| Parameters | <i>stringExp</i> is a string expression. It must be identical with an existing RAPID! program file name. |
| Description | <p>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.</p> <p>Global variables of the initially calling program can be used by all chained programs.</p> <p>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.</p> <p>Note: Subs and functions of the chained program will be ignored and not run if not explicitly called (see example below).</p> <p>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:</p> <ul style="list-style-type: none"> – 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. <p>A chained program may also chain other programs down to a depth of four nested CHAINS.</p> |

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 | <code>anyText</code> 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 | <pre>REM The next line will define a variable LET i = 42 ` Who does not call his integers "i"</pre> |

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 | <pre>PRINT "Please connect mobile now" WAIT 2000</pre> 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 | <p>Clears the screen and positions the cursor in the upper left screen corner.</p> <p>Optionally, new background and text settings for the subsequent output operations may be specified:</p> <p>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.</p> <p>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.</p> <p>The font setting is specified by the font code followed by the font size.</p> <p>The following fonts are available (corresponding codes in parentheses):</p> <p>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) .</p> <p>The selectable font sizes are 8, 10, 12, 14, 18, 20, 24, 32, 48 pt.</p> <p>Optionally, the text can be set to bold (b) and/or italic (i).</p> <p>Note: The font codes are string values, they have to be put in quotation marks (e.g. "cour14"). Otherwise, a syntax error will occur.</p> |

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 <i>x</i> is counted in character positions rather than in pixels; parameter <i>y</i> in screen lines. The values for <i>x</i> and <i>y</i> are dependent on the currently set font size. However, the 4400 enables the user to scroll the screen output. For practical reasons, <i>x</i> should be kept below 50 and <i>y</i> below 25. |
| Examples | <pre>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.)</pre> |

PRINT (OUTPUT)

| | |
|--------------------|--|
| Syntax | <pre>PRINT [{ [USING usestr\$,] exp [, ; TAB(number);] }]or OUTPUT [{ [USING usestr\$,] exp [, ; TAB(number);] }]</pre> |
| Parameters | <p><i>exp</i> is a numeric or string expression, <i>usestr\$</i> defines an output format for numeric data, <i>number</i> stands for an integer figure, forcing the next <i>exp</i> of the same PRINT or OUTPUT command to appear on a specific position on-screen.</p> |
| Description | <p>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 <i>a</i> gives the total amount of digits to be used for the output (including decimal places, the decimal point and the sign), <i>b</i> 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.</p> |

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 | <p>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).</p> <p>Note: There is no need to declare local variables. Within the RAPID! environment, all variables are interpreted as local by definition.</p> <p>However, after a variable has been declared using the DIM command, all subsequent variables have to be declared with DIM as well.</p> <p>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.</p> |
| 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 [{, parameter } ] ] )
  [ 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 | <p>Syntax 1</p> <pre>DO [{WHILE UNTIL} boolExp] [instructions] [EXIT DO] [instructions] LOOP</pre> <p>Syntax 2</p> <pre>DO [instructions] [EXIT DO] [instructions] LOOP [{WHILE UNTIL} boolExp]</pre> |
| Parameters | <p><code>boolExp</code> is a boolean expression, <code>instructions</code> are an (optional) number of program lines.</p> |
| Description | <p>Dependent on a condition (<code>boolExp</code>), this command structure repeatedly runs a program segment.</p> <p>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.</p> <p>Syntax 2 checks the condition after the loop has been run.</p> <p><code>WHILE</code> means that the loop will be run as long as <code>boolExp</code> is true.</p> <p><code>UNTIL</code> will run the loop until <code>boolExp</code> becomes true.</p> <p><code>EXIT DO</code> causes the loop to be left instantly.</p> <p>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 <code>EXIT DO</code> command somewhere within the loop to prevent the program from being trapped.</p> |

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 (`boolExp`), 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 $\neq 0$), 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. <pre>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</pre> |

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 212 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 operations. As long as a port has not been opened, I/O operations 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 positive 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 specify 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 listened 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 212 for details.
- `PRINT` also reads data from the SCPI system. Please check with section "[SCPI and RAPID!](#)" on page 269 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 | <code>BIN(string\$)</code> |
| Parameters | <code>string\$</code> is a string expression, containing a valid binary representation of an integer number. |
| Description | Returns the decimal representation of a binary expression contained in <code>string\$</code> . The value returned will be an integer. |
| Examples | <code>LET i=BIN("101")</code> assigns '5' to variable <code>i</code> . |

BIN\$

| | |
|-------------------|---|
| Syntax | <code>BIN\$(aNumber [, digits])</code> |
| Parameters | <code>aNumber</code> is a valid numeric expression, <code>digits</code> 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 | <code>OCT(string\$)</code> |
| Parameters | <code>string\$</code> is a string expression, containing a valid octal representation of an integer number. |
| Description | Returns the decimal representation of an octal expression contained in <code>string\$</code> . The value returned will be an integer number. |
| Examples | <code>LET i=OCT("15")</code> assigns '13' to variable <code>i</code> . |

OCT\$

| | |
|--------------------|--|
| Syntax | <code>OCT\$(aNumber [, digits])</code> |
| Parameters | <code>aNumber</code> is a valid numeric expression, <code>digits</code> is a numeric expression, resulting in a positive integer. |
| Description | Returns a string containing the octal representation of <code>aNumber</code> . In case, <code>aNumber</code> is not an integer, the decimals will be truncated (that means no rounding will take place). When <code>digits</code> is stated, the string will be filled up with zeros (0) to the left. |
| Examples | <code>LET a\$=OCT\$(254, 6)</code> assigns "000376" to variable <code>a\$</code> . |

VAL

| | |
|--------------------|---|
| Syntax | <code>VAL(string\$)</code> |
| Parameters | <code>string\$</code> 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 <code>string\$</code> to that decimal number. If the string does not contain a valid decimal number, '0' will be returned. |
| Examples | <code>i = VAL("1.234")</code> assigns '1.234' to variable <code>i</code> To avoid runtime errors due to typing errors with an <code>INPUT</code> command, first a string is read in that is later converted into the decimal representation of the figure entered. <code>INPUT "Please enter channel number:" _</code> <code>, channelno\$</code> <code>channelno = VAL(channelno\$)</code> |

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$, searchString$)`

Parameters `startPos` is a numeric expression, interpreted as positive integer.
`aString$` and `searchString$` are string expressions.

Description Searches `aString$` for the first occurrence of `searchString$`.
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$, "national")`
`j = INSTR(8, a$, "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-yyyy", where "mmm" represent a three-letter abbreviation of the month, "dd" stand for the integer, giving the day of the month and "yyyy" 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 fileHandle, 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 following 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 creating an event; time simply expired without anything happening. 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 | <code>path\$</code> is a valid string expression. |
| Description | Creates a new directory with the <code>path\$</code> specified. <code>path\$</code> may either be a relative or absolute directory name. If <code>path\$</code> is not a valid directory name, a runtime error will occur. |
| Examples | <code>MKDIR "/usr/new"</code> |

NAME

| | |
|--------------------|---|
| Syntax | <code>NAME oldName\$ AS newName\$</code> |
| Parameters | <code>oldName\$</code> and <code>newName\$</code> are both valid string expressions. |
| Description | Changes the name of the file specified with <code>oldName\$</code> to <code>newName\$</code> . Both <code>oldName\$</code> and <code>newName\$</code> have to be valid file names. If <code>oldName\$</code> is currently open, a runtime error will occur. |
| Example | <code>NAME "/usr/new/file.txt" AS "/usr/new/file_2.txt"</code> |

RMDIR

| | |
|--------------------|--|
| Syntax | <code>RMDIR path\$</code> |
| Parameters | <code>path\$</code> is a valid string expression. |
| Description | Deletes a directory with the <code>path\$</code> specified without prompting. <code>path\$</code> may be either a relative or an absolute directory name. If <code>path\$</code> is not a valid directory name or if the directory is not empty, a runtime error will occur. |
| Example | <code>RMDIR "/usr/new"</code> |

SHELL

| | |
|-------------------|--|
| Syntax | <code>SHELL parameter\$</code> |
| Parameters | <code>parameter\$</code> 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 259](#) 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, respectively. |
| 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 | <p>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).</p> <p>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).</p> <p>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).</p> <p>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).</p> |
| 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 '&t', 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 | <p>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.</p> |
| 101 | <p>Too many GOSUBs Overflow of the GOSUB . . . RETURN stack. To avoid this error message, reduce the number of recursive procedure calls.</p> |
| 102 | <p>RETURN without GOSUB A RETURN command was found, but the program is in no procedure currently.</p> |
| 103 | <p>VAL: Number expected The VAL function returns a numeric expression (contained in a string expression).</p> |
| 104 | <p>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.</p> |
| 105, 106 | <p>Index out of range The index of an indexed variable must be in the range of -32,766 to +32,766.</p> |
| 107 | <p>ON ERROR not allowed in an error handling routine Within program-internal error handling, an ONERROR command is not allowed.</p> |
| 109 | <p>Dimension Error The physical dimensions of the operands do not match (e.g. $i = 1 \text{ kHz} + 5 \text{ mW}$).</p> |
| 110 | <p>No such file or directory The file name or directory specified does not exist.</p> |
| 111 | <p>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 longer required to be open.</p> |
| 112 | <p>Permission denied There is no permission to access the specified file or directory.</p> |
| 113 | <p>Invalid access code The access code used is not valid and access is still denied.</p> |

| | |
|-----|---|
| 114 | <p>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.</p> |
| 115 | <p>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.</p> |
| 116 | <p>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.</p> |
| 117 | <p>File already exists The file or directory specified does exist and cannot be created anew.</p> |
| 118 | <p>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'.</p> |
| 119 | <p>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.</p> |
| 120 | <p>Directory busy The directory specified is currently in use by another task and cannot be accessed.</p> |
| 121 | <p>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.</p> |
| 122 | <p>Filename must not be empty The string specifying a file name is empty.</p> |
| 123 | <p>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.</p> |
| 124 | <p>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.</p> |

| | |
|-----|--|
| 125 | <p>Open GPIB failed The GPIB port could not be opened. Please check whether all parameters used in the OPEN "GPIB0:..." command are correct.</p> |
| 126 | <p>SCPI already open The SCPI port may only be opened once.</p> |
| 127 | <p>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.</p> |
| 128 | <p>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.</p> |
| 199 | <p>USER BREAK The program was terminated by the user.</p> |
| 201 | <p>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.</p> |
| 202 | <p>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.</p> |
| 203 | <p>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.</p> |

SCPI

8

This chapter describes the remote control capabilities of the 4400. Test results described in this chapter are as follows:

- ["Overview" on page 264,](#)
- ["What SCPI is" on page 264](#)
- ["Structure" on page 265](#)
- ["Syntax and notation" on page 265](#)
- ["SCPI and RAPID!" on page 269](#)
- ["Command subsystem overview" on page 276](#)
- ["The Node B and UE parameter subsystems" on page 286](#)
- ["The communication-related subsystems" on page 278](#)
- ["SCPI command errors" on page 305](#)

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 (Standard Commands for Programmable Instruments) 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 271](#).

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 267.

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 273.

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 1a 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 279](#). The use of the service register is outlined in subsection ["At the top: the service register" on page 281](#).

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 272](#) 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 273](#)
- ["Message exchange" on page 274](#)
- ["Example of a GPIB protocol" on page 275](#)

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 cali- bration 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 cali- bration |
| from | :PROG:MESS "RX, 1, 62, 124, 15" | Start (user-programmed) RX cali- bration 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 cali- bration |
| 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

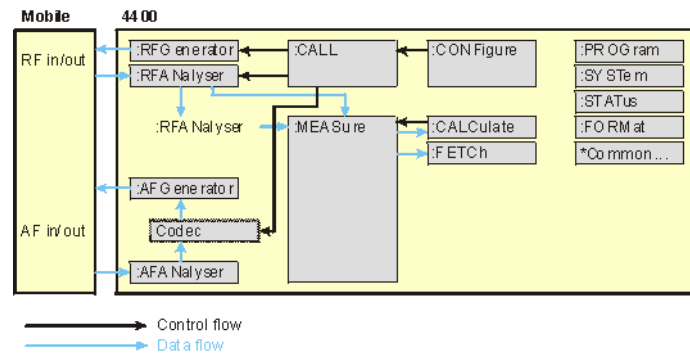
[SCPI command errors](#)

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 or 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 GPIB 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 279](#) before using the status subsystem.

A table of all registers implemented can be found in subsection ["Table of registers" on page 282](#).

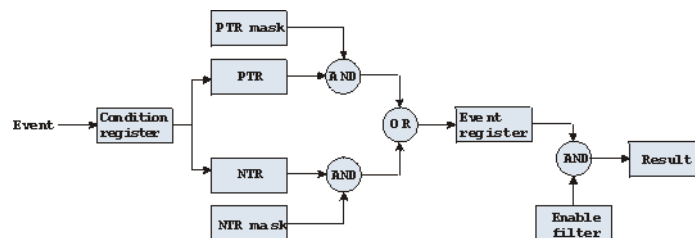
Understanding the STATus subsystem

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 reflect 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 Positive TRansition and the Negative TRansition 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 RFSPpectrum 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:CSYSstem | 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 latest 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 measurement | 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 | <code>:MEASure[:CONTInuous]<{:measProp}>[?]</code> |
| Parameters | The one-shot measurements do not require any parameters. |
| Description | <p>Starts the (continuous) measurement of the specified measurable property. The <code>CONTInuous</code> command element is optional.</p> <p>Unless a measurement result is read out (using the <code>FETCh</code> subsystem), no measurement result values will be given back.</p> <p>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.</p> <p>The latest result measured may be read out using the <code>FETCh</code> subsystem.</p> <p>Any <code>FETCh</code> 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 <code>FETCh</code> 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 <code>STATus</code> subsystem will be set, too.</p> <p>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.</p> <p>The main application of the combination of the <code>MEAS</code> and the <code>FETC</code> 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 <code>MEAS</code> command, the test set 'is free' to perform other tasks. If in this case the query form of the <code>MEAS</code> command is used, the test set is blocked until a measurement result is available.</p> <p>Sometimes, measurement results will need to 'sweep in'. In this case, the first measurement result might be totally misleading. Using the <code>MEAS</code> command will allow the measurement result value to stabilize on a meaningful result before the latest result value is read out using the <code>FETC</code> subsystem.</p> |
| Query | <p>The query form of any <code>MEASure</code> command will start the (continuous) measurement of the specified measurable property.</p> <p>The <code>CONTInuous</code> command element is optional.</p> <p>After the first measurement has been completed, the measurement result value(s) will be delivered back in a string as outlined above.</p> <p>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.</p> <p>The main application of the query form of a <code>MEAS</code> command is speed. When combined with 'fast' measurements (like e.g. the fast power level measurement), the query form of a <code>MEAS</code> command delivers measurement results as fast as possible.</p> <p>Note: The measurement started with the query form of the <code>MEAS</code> command will be continued in the background. Any further measurement result values may be read out, using the appropriate <code>FETCh</code> command.</p> |

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 287 for details).

| | |
|--------------------|---|
| Syntax | MEASure:ARRay<{ :measProp }>[?] <numMeas> |
| Parameters | numMeas is the number of measurements to be performed. |
| Description | <p>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).</p> <p>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 289).</p> |
| Query | <p>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.</p> <p>The query will then return a string containing all the measurement results. The single result values will be separated by commas.</p> <p>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 289).</p> |
| Examples | <pre>:MEASure:GSM:ARRay:RFTX:PPEAK 10</pre> <p>This command will make the 4400 perform 10 independent measurements of the maximum phase error in GSM.</p> <p>After those 10 results have been achieved, the measurement will be stopped. The ten result values will be stored in an internal array.</p> <p>To read out the measurement result array, use the FETCh subsystem:</p> <pre>:FETCh:GSM:RFTX:PPEAK?</pre> <p>will return the 10 values in one string (like "5.42,5.44,5.80,...5.72,5.64")</p> <p>Any subsequent :FETCh:GSM:RFTX:PPEAK? command will result in a timeout.</p> <pre>MEASure:GSM:ARRay:RFTX:ALL? 2</pre> <p>This command takes all relevant RFTX measurements twice. The measurements will be stopped as soon as the 2 x 19 result values are available.</p> <p>The 38 result values will be returned as a string; the single values will be separated by commas.</p> <p>Any subsequent :FETCh:GSM:RFTX:PPEAK? command will result in a timeout.</p> |

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:PHASerror

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:PHASerror` 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:MSPpectrum` 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 `:RFSPpectrum:MSPpectrum:SPAN` and `:RFSPpectrum:MSPpectrum:RESolution`.

:MEASure:BLOCKdata:MSPpectrum: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:MSPpectrum` 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 `:RFSPpectrum:MSPpectrum:SPAN` and `:RFSPpectrum:MSPpectrum:RESolution`.

:MEASure:BLOCKdata:AFSPpectrum

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:AFSPpectrum` 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 ["The MEASure subsystem" on page 287](#)), 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, FREQUency, POWer, LENGth :MEAS:RFTX:GRO This sequence of commands first defines a group of RFTX measurements and then issues a group command. As soon as all the measurements specified in the group command have been completed, the measurement result values can be read out using the :FETCh:GSM:RFTX:GRO? command. |

The FETCh subsystem

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

Important notes

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

There are two versions of a FETCh command:

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

FETCh:LAST

The **:FETCh:LAST?** command will read out the latest result of the last MEASurement command issued - whatever command that was. Using this command, please keep in mind that your control program then has to take care of the number and format of the measurement result values returned. To convert a result string back into single measurement result values, please refer to section ["How to convert a result string back into numeric variables" on page 290](#).

| | |
|--------------------|---|
| 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.

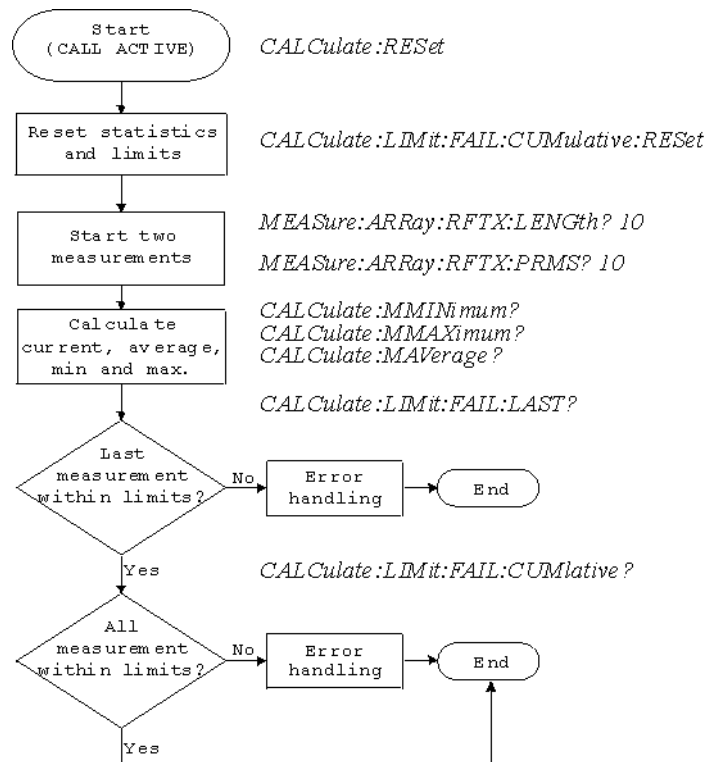
The CALCulate Subsystem

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).



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 299.
- 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 | <p>:MEAS:GSM:ARR:RFTX:LENG :MEAS:GSM:ARR:RFTX:PRMS :CALC:GSM:LIM:FAIL:LAST?</p> <p>This command of the CALCulate subsystem will check whether the latest measurement result value of the PRMS measurement is within its limits.</p> <p>:MEAS:GSM:ARR:RFTX:LENG 10 :CALC:GSM:LIM:FAIL:LAST?</p> <p>In this example, the :CALC command will check if all 10 measurement result values of the burst length measurement array are within their limits.</p> |

{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 | <p>Checks whether measurement result values of the measurement specified with the {measProp} command element are within their limits.</p> <p>In case, this command is used during a continuous measurement, only the latest measurement result will be checked.</p> <p>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.</p> |

| | |
|----------------|---|
| Example | <pre> :MEAS:GSM:ARR:RFTX:LENG 10 :MEAS:GSM:ARR:RFTX:PRMS 10 :CALC:GSM:RFTX:LENG:LIM:FAIL? </pre> <p>In this example, the <code>:CALC</code> command will check whether all 10 measurement result values of the <code>LENG</code> measurement array are within their limits.</p> |
|----------------|---|

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

| | |
|--------------------|---|
| Syntax | <code>CALCulate:LIMit:FAIL:CUMulative?</code> |
| 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 | <pre> :MEAS:GSM:ARR:RFTX:LENG 10 :MEAS:GSM:ARR:RFTX:PRMS 10 :CALC:GSM:LIM:FAIL:CUM? </pre> <p>This command of the <code>CALCulate</code> subsystem will check whether all measurement result values of the <code>PRMS</code> measurement are within their limits.</p> |

FAIL:CUM:RESet – resets the cumulative limit evaluation

| | |
|--------------------|---|
| Syntax | <code>CALCulate:LIMit:FAIL:CUMulative:RESet</code> |
| Description | Resets (clears) the cumulative check of measurement result values against their limits. Only the measurement results from <code>:MEAS:...</code> 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 | <code>CALCulate:{measProp}:LIMit:STATe</code> |
| Description | Switches the limit check for a measurement specified with the <code>{measProp}</code> command element either ON or OFF . |
| Example | <pre> :CALC:RFTX:ALL:LIM:STAT OFF </pre> <p>Switches the limit check for the 19 main RF TX measurements off.</p> |

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

| | |
|--------------------|--|
| Syntax | <code>CALCulate:{measProp}:LIMit:UPPer[:DATA]</code> |
| Description | Sets the upper limit for the limit check of the measurement specified with the <code>{measProp}</code> ; 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 | <p>: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.</p> |
|----------------|--|

:{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 | <p>: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.</p> |

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 | <p>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.</p> |

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:STATe** 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 303 . |
| 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 returned 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 measurement result is beyond the limits, a 1 (fail) will be returned. |
| 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 evaluation 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 minimum 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.

RFSPpectrum – 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 functionality 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 286](#).
- 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



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" on page 310](#)
- ["Common commands" on page 311](#)
- ["SYSTem subsystem" on page 316](#)
- ["STATus subsystem" on page 324](#)
- ["PROGram subsystem" on page 340](#)
- ["FORMat subsystem" on page 343](#)
- ["CONFigure subsystem" on page 345](#)
- ["CALL subsystem" on page 369](#)
- ["MEASure subsystem" on page 377](#)
- ["FETCh Subsystem" on page 417](#)
- ["CALCulate Subsystem" on page 438](#)
- ["READ subsystem" on page 503](#)
- ["RFANalyser subsystem" on page 516](#)
- ["RFSPectrum subsystem" on page 517](#)
- ["AFGenerator subsystem" on page 519](#)
- ["AFANalyser subsystem" on page 523](#)
- ["MS Power Supply subsystem" on page 527](#)

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: <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - manufacturer's name - name of the device - serial number - software revision number <p>All parameters are separated by commas.</p> <p>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.</p> |
| 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 | <p>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.</p> |
| Description | <p>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.</p> |
| Query | <p>Reads and returns the current setting for the printer. The string returned will contain one of the predefined expressions as explained above.</p> |
| Example | <p>:SYST:PRIN? String returned: "HPLJ".</p> |

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 279.

: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 | <p>There is only a query form of this command available. The query will return a string, containing one integer.</p> <p>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.</p> <p>Notes</p> <ul style="list-style-type: none"> - For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. - In case, a certain event shall be trapped in a loop within a program, always query the event-type register. |
| Example | <p>:STATus:OPERation:EVENT? Value returned: "32".</p> <p>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).</p> |

:STATus:OPERation:CONDition?

| | |
|--------------------|---|
| Syntax | :STATus:OPERation:CONDition? |
| Parameters | There are no parameters. |
| Description | <p>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.</p> <p>Note: This register is nondestructive. This means that it will keep its contents after any query.</p> |
| Query | <p>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.</p> <p>Notes</p> <ul style="list-style-type: none"> - For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. - Bit 15 (the MSB) of this register is not used. Therefore, the maximum value returned will be "32767". |
| Example | <p>:STATus:OPERation:CONDition? Value returned: "512".</p> <p>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.</p> |

: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 ANDed 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 ANDed 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 <ul style="list-style-type: none"> - 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 ANDed 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 <ul style="list-style-type: none"> - 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:COMPLete:CONDition

| | |
|--------------------|--|
| Syntax | :STATus:OPERation:COMPLete:CONDition? |
| Parameters | There are no parameters. |
| Description | There is only a query form available. |
| Query | Returns a string containing an integer number (for CDMA operation only). The number indicates which condition(s) lead to completion. Valid numbers are: 1 – indicating that the signaling has been completed. |
| Example | :STATus:OPERation:COMP:COND? Results in a string like this: 1, meaning that the signaling operation is completed. |

:STATus:OPERation:COMPLete:ERRor

| | |
|--------------------|---|
| Syntax | :STATus:OPERation:COMPLete:ERRor? |
| Parameters | There are no parameters. |
| Description | There is only a query form available. |
| Query | The query returns TRUE if an error occurred, otherwise it returns FALSE (for CDMA operation only). |
| Example | :STAT:OPER:COMP:ENAB 1 :CALL:CDMA:REG;*WAI; This will cause the 4400 to wait until the phone has registered or an error has occurred. :STAT:OPER:COMP:ERR? To check if an error has occurred. |

:STATus:OPERation:COMPLete:ENABle

| | |
|--------------------|--|
| Syntax | :STATus:OPERation:COMPLete:ENABle <int1> |
| Parameters | int1 is an integer; the valid range for int1 is from 0 to 32767 and the default value is 0. |
| Description | This command sets the conditions for which to wait until completion (for CDMA operation only). A value of 1 enables waiting for call processing to complete. |
| Query | There is no query form available. |
| Example | :STAT:OPER:COMP:ENAB 1 This command enables the signaling complete check. :CALL:CDMA:REG;*WAI; This will cause the 4400 to wait until the phone has registered or an error has occurred. :STAT:OPER:COMP:ERR? To check if an error has occurred. |

:STATus:OPERation:SIGNalling:CDMA[:EVENT]?

| | |
|-------------------|--|
| Syntax | :STATus:OPERation:SIGNalling:CDMA[:EVENT]? |
| Parameters | There are no parameters. |

| | |
|--------------------|---|
| Description | <p>Reads out the current contents of the signaling operation event register.</p> <p>Note: Event-type registers are read-only and self-destructive. They will be cleared after any query.</p> |
| Query | <p>There is only a query form of this command available. The query will return a string containing one integer.</p> <p>The value returned represents all signaling operation events that have occurred since the last query of this register. As with any event-type register, the individual bits will remain set even when the reason for the bits to be set is no longer valid. Please note that this is the main difference between event-type and condition-type registers. Condition-type registers reflect the current state of Willtek 4400. Consequently, condition-type registers will be updated continuously.</p> <p>The bits of the signaling operation event register have the following meaning:</p> <p>Bit 0 (decimal 1): CDMA mode disabled.</p> <p>Bit 1 (decimal 2): The 4400 is in CDMA mode and on a control channel.</p> <p>Bit 2 (decimal 4): Alerting mobile, waiting for the phone to answer the call.</p> <p>Bit 3 (decimal 8): 4400 and phone on a traffic channel.</p> <p>Bit 4 (decimal 16): Async mode (i.e. generator/analyzer) active.</p> <p>Bits 5 to 7: Not used (reserved).</p> <p>Bit 8 (decimal 256): Mobile registration received.</p> <p>Bit 9 (decimal 512): Beginning a BS-initiated call.</p> <p>Bit 10 (decimal 1024): Beginning a MS-initiated call.</p> <p>Bit 11 (decimal 2048): RESERVED – DTMF key press detected.</p> <p>Bit 12 (decimal 4096): Mobile power report received.</p> <p>Bit 13 (decimal 8192): Mobile pilot strength report received.</p> <p>Bit 14 (decimal 16384): Mobile power correction completed (if :CONF:CDMA:BS:TCH:PCON:RES was issued, the flag indicates that the Power Control Reset has finished).</p> <p>Bit 15 (decimal 32768): Not used (reserved).</p> <p>Notes:</p> <ul style="list-style-type: none"> – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. – In case a certain event shall be trapped in a loop within a program, always query the event-type register. |
| Example | <p><code>:STATus:OPERation:SIGNalling:CDMA:EVENT?</code></p> <p>Value returned: "8".</p> <p>This means that bit 3 signaling operation event register has been set. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 310.</p> |

:STAT:OPERation:SIGNalling:CDMA:CONDition?

| | |
|--------------------|--|
| Syntax | <code>:STAT:OPERation:SIGNalling:CDMA:CONDition?</code> |
| Parameters | There are no parameters. |
| Description | <p>This command reads out the current contents of the signaling operation condition register. This register reflects the current state of Willtek 4400 in terms of signaling and will be updated continuously.</p> <p>Note: This register is nondestructive. This means that it will keep its contents after any query.</p> |

| | |
|----------------|--|
| Query | <p>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.</p> <p>Note: For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279.</p> |
| Example | <pre>:STATus:OPERation:SIGNalling:CDMA:CONDition?</pre> <p>Value returned: "512".</p> <p>This means that bit 9 has been set. Bit 9 will be reset as soon as the change of the call channel has been completed. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 310.</p> |

| Bit | Decimal | Meaning |
|-----|---------|---|
| 0 | 1 | CDMA mode disabled. |
| 1 | 2 | The 4400 is in CDMA mode and on a control channel. |
| 2 | 4 | Alerting mobile, waiting for the phone to answer the call. |
| 3 | 8 | 4400 and phone on a traffic channel. |
| 4 | 16 | Async mode (i.e. generator/analyzer) active. |
| 5 | 32 | Not used (reserved). |
| 6 | 64 | Not used (reserved). |
| 7 | 128 | Not used (reserved). |
| 8 | 256 | Mobile registration received. |
| 9 | 512 | Beginning a BS-initiated call. |
| 10 | 1024 | Beginning a MS-initiated call. |
| 11 | 2048 | RESERVED – DTMF key press detected. |
| 12 | 4096 | Mobile power report received. |
| 13 | 8192 | Mobile pilot strength report received. |
| 14 | 16384 | Mobile power correction completed (if :CONF:CDMA:BS:TCH:PCON:RES was issued, the flag indicates that the Power Control Reset has finished). |
| 15 | 32768 | Not used (reserved). |

:STATus:OPERation:SIGNalling:CDMA:ENABLe

| | |
|-------------------|--|
| Syntax | <pre>:STATus:OPERation:SIGNalling:CDMA:ENABLe <int1></pre> |
| Parameters | <pre>int1</pre> is an integer. The minimum value for <pre>int1</pre> 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 ANDed with the signaling operation event register and thus decide what kind of events will be forwarded to bit 10 of the general operation register group. |
| Query | There is no query form of this command available. |
| Example | <code>:STATus:OPERation:SIGNalling:CDMA:ENABle 16</code> This means that any setting of bit 4 of the signaling operation event register will rise bit 8 of the general operation condition register. For details on return data (bits and decimals and their meaning) refer to section "Signaling operation register group" on page 310. |

:STAT:OPERation:SIGNalling:CDMA:PTRansition

| | |
|--------------------|--|
| Syntax | <code>:STAT:OPERation:SIGNalling:CDMA:PTRansition <int1></code> |
| Parameters | <code>int1</code> is an integer. The minimum value for <code>int1</code> 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 ANDed with the up to 15 bits of the PTR filter of the signaling operation condition register to allow a positive transition of any bit (i.e. a transition from logic '0' to '1') to reach the signaling operation event register. Note: The default of this mask is 32767 – that means that the mask will allow any positive transition of the lower 15 bits of the condition-type register to reach the event-type register. |
| Query | There is no query form of this command available. |
| Example | <code>:STATus:OPERation:SIGNalling:CDMA:PTRansition 0</code> 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 310. |

:STAT:OPERation:SIGNalling:CDMA:NTRansition

| | |
|--------------------|---|
| Syntax | <code>:STAT:OPERation:SIGNalling:CDMA:NTRansition <int1></code> |
| Parameters | <code>int1</code> is an integer. The minimum value for <code>int1</code> 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 ANDed 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 | <code>:STATus:OPERation:SIGNalling:CDMA:NTRansition 0</code> 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 310. |

: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 <ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> - For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. - 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 ANDed 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 ANDed 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 ANDed 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 <ul style="list-style-type: none"> - For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. - 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 <ul style="list-style-type: none"> - For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. - 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 ANDed 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 ANDed 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 ANDed 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 <ul style="list-style-type: none"> – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. – 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 |
| | <ul style="list-style-type: none"> – For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. – Bits 4 to 15 (the MSB) of this register are not used. Therefore, the maximum value returned will be "15" |
| Example | <p>: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.</p> |

: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 ANDed 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 | <p>: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.</p> |

: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 | <p>Sets the PTR mask of the "RF questionable status register group". This mask will be ANDed 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.</p> <p>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.</p> |
| Query | There is no query form of this command available. |
| Example | <p>: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.</p> |

: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 ANDed 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 <ul style="list-style-type: none"> - For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. - 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 <ul style="list-style-type: none"> - For further details regarding the basic functions of the STATus subsystem, please refer to section "Understanding the STATus subsystem" on page 279. - 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 ANDed 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 ANDed 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 ANDed 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. |

PROG

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

:PROGram[**:SE**Lected] :NAME

| | |
|--------------------|---|
| Syntax | :PROG |
| 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 :PROG ram[:SE Lected] :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 183. |
| Query | The query returns the name of the currently loaded RAPID! basic program file as a string. |
| Example | :PROG ram:SE |

:PROGram[**:SE**Lected] :NUM

| | |
|--------------------|--|
| Syntax | :PROG |
| 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 int x is -32765, the maximum value is 32765. The default value for each int x 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 204. |
| Query | The query returns the current value of a numerical variable as one floating point real number contained in a string. |
| Example | :PROG ram:SE |

: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 204 . |
| 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 183 . |
| 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 | :PROG | ram[: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 | :PROG ram:SELEcted:MESSAge "Put this in the queue" :PROG :MESS? | The result read back will be "Put this in the queue". |

:PROGram[:SELEcted]:SRE

| | | |
|--------------------|----------------------------------|--|
| Syntax | :PROG | ram[: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 279. |
| Query | | There is no query form of this command available. |
| Example | :PROG ram:SELEcted:SRE 16 | This will set bit 4 of the mask of the service register. The contents of the service register will be ANDed 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 | :PROG | ram[: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 279. |
| Query | | The command returns a string containing one integer. This integer represents the contents of the service register of the RAPID! program. |
| Example | :PROG ram: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 | <code>:FORMat:MRESult:STYPe <PredefExp></code> |
| Parameters | PredefExp is one of the following predefined expressions: STB SIGNaling 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". SIGNaling 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 | <code>:FORMat:MRESult:HEADer <PredefExp></code> |
|---------------|---|

| | |
|--------------------|---|
| 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:CSYSstem | 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:CSYSstem` command first before you change any parameters using different SCPI commands.

:CONFigure:CSYSstem

Syntax `:CONFigure:CSYSstem <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 | <p>Selects the type of communication system to be used.</p> <p>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.</p> <p>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.</p> <p>GCGenana will set up the Willtek 4400 as a generator and analyzer for circuit-switched GSM signals (including multislots/HSCSD). Generator/analyzer means that there will be no call setup and therefore, all measurements will be asynchronous.</p> <p>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 multislots mode as well as packet-data channels (PDTCH in GPRS)).</p> <p>GSM will set up the Willtek 4400 as a test set for circuit-switched GSM systems (including multislots/HSCSD). All tests performed with this setting require a call setup. Therefore, this test mode is called the 'call mode'.</p> <p>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 multislots mode as well as packet-data channels (GPRS)).</p> <p>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.</p> |
| 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 | <pre>:CONFigure:CSYSTEM GSM :CONF:CSYS? Value returned: "GSM".</pre> |

:CONFigure:COUPloss:STATE

| | |
|--------------------|---|
| Syntax | :CONFigure:COUPloss:STATE <PredefExp1> |
| Parameters | PredefExp1 is one of the following predefined expressions: ON OFF . Default is OFF . |
| Description | <p>This command switches the use of a coupling loss table either on or off.</p> <p>Note: Coupling loss tables are used to compensate e.g. losses in cables. For more details, please refer to section Coupling Loss.</p> |
| 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 | <pre>:CONFigure:COUPloss:STATE ON :CONF:COUP:STAT? Value returned: "ON"</pre> |

:CONFigure:COUPloss:NAME

| | |
|--------------------|---|
| Syntax | :CONFigure:COUPloss:NAME <string1> |
| Parameters | string1 is a string, giving the complete file name (and directory) of the coupling loss file to be loaded. The maximum length of string1 is 50 characters. The default for string1 is "example.cpl". |
| Description | This command loads the coupling loss description file, specified with the command's parameter. Please note that the data contained in the file need to be activated (using the CONF:COUP:STAT ON command described above) before the data contained in the file specified will have any effect on the measurement results. For more details, please refer to section "Coupling Loss". |
| Query | The query form of this command will return the name of the currently loaded coupling loss description file. The string delivered back will contain the full file name. |
| Example | :CONFigure:COUPloss:NAME "m7389.cpl" :CONF:COUP:NAME? Value returned: "m7389.cpl" |

:CONFigure:COUPloss:INFormation?

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

:CONFigure:COUPloss:DATA

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

| | |
|--------------------|---|
| Parameters | <p>string1 is a comment line related to the coupling loss data. The maximum length of string1 is 255 characters.</p> <p>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.</p> <p>The minimum resolution for all realfx values is 10 Hz (0.00001 MHz).</p> <p>The default value for all realfx is 800.0 MHz.</p> <p>The minimum value for all realax is -5.0 dB.</p> <p>The maximum value for all realax is 40.0 dB.</p> <p>The minimum resolution for all realax is 0.01 dB.</p> <p>The default value for all realax is 0.0.</p> <p>Notes</p> <ul style="list-style-type: none"> - 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 | <code>:CONFigure:COUPlOSS:DATA "Motorola 7389 with cable #23",825.0,15.0,1750.0,19.0</code> |

:CONFigure:ESYNc

| | |
|--------------------|--|
| Syntax | <code>:CONFigure:ESYNc?</code> |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | <p>The query form will return the status of the current external synchronization. The string delivered back will contain one of the following expressions:</p> <p>NONE MHZ5 MHZ10 MHZ13.</p> |
| Example | <p><code>:CONFigure:ESYNc?</code></p> <p>Value returned for example: "MHZ10".</p> <p>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.</p> |

:CONFigure:CDMA:REFErence:CLOCK

| | |
|--------------------|---|
| Syntax | <code>:CONFigure:CDMA:REFErence:CLOCK <PredefExp></code> |
| Parameters | <p>PredefExp is one of the following predefined expressions:</p> <p>CHIPx1 CHIPx16 CHIPx8 EVEN PCG PN FRAMe SFRAMe. Default is CHIPx1.</p> |
| Description | This command sets the CDMA reference clock output signal. |

| | |
|----------------|---|
| Query | The query form of this command returns the current setting for <code>PredefExp</code> . |
| Example | <code>:CONF:CDMA:REF:CLOC CHIPx16</code> <code>:CONF:CDMA:REF:CLOC?</code> Value returned: <code>CHIPx16</code> . |

:CONFigure:CDMA:BS:LEVel

| | |
|--------------------|---|
| Syntax | <code>:CONFigure:CDMA:BS:LEVel <Real1></code> |
| Parameters | <code>Real1</code> is a floating point real number. The minimum value is <code>-120.0</code> , the maximum value is <code>0.0</code> . The minimum resolution is <code>0.1</code> . The default value is <code>-60.0</code> . |
| Description | This command sets the RF output power level of the Willtek 4400 (i.e. simulated base station). The physical dimension is dBm. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one real number. |
| Example | <code>:CONF:CDMA:BS:LEVel -50.5</code> <code>:CONF:CDMA:BS:LEV?</code> Value returned: <code>-50.5</code> . |

:CONFigure:CDMA:BS:PAGing

| | |
|--------------------|---|
| Syntax | <code>:CONFigure:CDMA:BS:PAGing <PredefExp></code> |
| Parameters | <code>PredefExp</code> is one of the following predefined expressions: <code>FULL</code> <code>HALF</code> . Default is <code>FULL</code> . |
| Description | This command sets the paging rate. |
| Query | The query form of this command returns the current setting for <code>PredefExp</code> . |
| Example | <code>:CONF:CDMA:BS:PAG FULL</code> <code>:CONF:CDMA:BS:PAG?</code> Value returned: <code>FULL</code> . |

:CONFigure:CDMA:BS:PNOFfset

| | |
|--------------------|--|
| Syntax | <code>:CONFigure:CDMA:BS:PNOFfset <Int1></code> |
| Parameters | <code>Int1</code> is an integer number. The minimum value is <code>0</code> , the maximum value <code>511</code> . The default value is <code>0</code> . |
| Description | This command sets the pilot PN offset. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | <code>:CONF:CDMA:BS:PNOF 0</code> <code>:CONF:CDMA:BS:PNOF?</code> Value returned: <code>0</code> . |

:CONFigure:CDMA:BS:MCC

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:MCC <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 999. The default value is 1. |
| Description | This command sets the mobile country code (MCC). |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:MCC 1 :CONF:CDMA:BS:MCC? Value returned: 1. |

:CONFigure:CDMA:BS:MNC

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:MNC <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 99. The default value is 0. |
| Description | This command sets the mobile network code (MNC). |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:MNC 0 :CONF:CDMA:BS:MNC? Value returned: 0. |

:CONFigure:CDMA:BS:SYNC:SID

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:SYNC:SID <Int1> |
| Parameters | Int1 is an integer number in the range from 0 to 32767. The default is 1. |
| Description | This command sets the system identification (SID). |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:SYNC:SID 1 :CONF:CDMA:BS:SYNC:SID? Value returned: 1. |

:CONFigure:CDMA:BS:SYNC:NID

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:SYNC:NID <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 65535. The default value is 0. |
| Description | This command sets the network identification (NID). |

| | |
|----------------|--|
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:SYNC:NID 0 :CONF:CDMA:BS:SYNC:NID? Value returned: 0. |

:CONFigure:CDMA:BS:CCH:BAND

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:CCH:BAND <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. Default is US800. |
| Description | This command sets the control channel band. |
| Query | The query form of this command returns the current setting. |
| Example | :CONF:CDMA:BS:CCH:BAND US800 :CONF:CDMA:BS:CCH:BAND? Value returned: US800. |

:CONFigure:CDMA:BS:CCH:ARFCn

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:CCH:ARFCn <Int1>, [<PredefExp>] |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 2047. The default value is 29. PredefExp is an optional expression which can take on one of the following values: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. The default is US800. |
| Description | This command sets the channel for the control channel. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:BS:CCH:ARFC 0 :CONF:CDMA:BS:CCH:ARFC? Value returned: 0. |

:CONFigure:CDMA:BS:TCH:BAND

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:TCH:BAND <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. Default is US800. |
| Description | This sets the traffic channel band. |
| Query | The query form of this command returns the selected band. |
| Example | :CONF:CDMA:BS:TCH:BAND US800 :CONF:CDMA:BS:TCH:BAND? Value returned: US800. |

:CONFigure:CDMA:BS:TCH:ARFCn

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:ARFCn <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 2047. The default value is 29. |
| Description | This command sets the channel number for the traffic channel. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:TCH:ARFC 0 :CONF:CDMA:BS:TCH:ARFC? Value returned: 0. |

:CONFigure:CDMA:BS:TCH:RCONfig:[FORWARD]

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:RCONfig:[FORWARD] <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 1, the maximum value 5. The default value is 1. |
| Description | This command sets the forward radio configuration. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:TCH:RCON:FORW 3 :CONF:CDMA:BS:TCH:RCON:FORW? Value returned: 3. |

:CONFigure:CDMA:BS:TCH:RCONfig:ACTiv

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:RCONfig:ACTive |
| Parameters | There are no parameters. |
| Description | This command returns the forward radio configuration, which may be different from the default forward radio configuration. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:TCH:RCON:ACT? Value returned: 3. |

:CONFigure:CDMA:BS:TCH:SOPTion:[DATA]

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:TCH:SOPTion:[DATA] <Int1>, <Int2> |
| Parameters | Int1 is an integer value with a minimum value of 1, the maximum value is 32768. The default value is 1. Int2 is an integer value. The minimum value for Int2 is 1, the maximum value 5, the default value is equal to the current forward radio configuration. |
| Description | This command sets the default service options for a particular radio configuration. |

| | |
|----------------|--|
| Query | The query form of this command will return the current setting for a particular radio configuration. |
| Example | :CONF:CDMA:BS:TCH:SOPT 2,3 :CONF:CDMA:BS:TCH:SOPT? 3 Value returned: 2. |

:CONFigure:CDMA:BS:TCH:SOPTion:ACTive

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:SOPTion:ACTive |
| 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 current setting of the Service Option which maybe different from the default value for a particular radio configuration once a call has been established. |
| Example | :CONF:CDMA:BS:TCH:SOPT:ACT? Value returned: 2. |

:CONFigure:CDMA:BS:TCH:PCONTrol:SElect

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:TCH:PCONTrol:SElect <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ALternating ACTive UP DOWN. Default is ALternating. |
| Description | This sets how power control should be handled with the mobile. |
| Query | The query form of this command returns the current setting. |
| Example | :CONF:CDMA:BS:TCH:PCON:SEL ACT :CONF:CDMA:BS:TCH:PCON:SEL? Value returned: ACT. |

:CONFigure:CDMA:BS:TCH:PCONTrol:RESet:[IMMediate]

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:TCH:PCONTrol:RESet:[IMMediate] |
| Parameters | No parameters. |
| Description | This performs a power control reset. This is only valid when the power control mode selected is ALternating, which returns the mobile power to initial test conditions. |
| Query | No query is available with this command. |
| Example | :CONF:CDMA:BS:TCH:PCON:RES:IMM |

:CONFigure:CDMA:BS:TCH:PCONtrol:ONCE

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:PCONtrol:ONCE <PredefExp> , <Int1> |
| Parameters | PredefExp is one of the following predefined expressions: UP DOWN. Int1 is an integer number with a minimum value of 1, the maximum value is 100. The default value is 1. |
| Description | This command causes a number of power control bits to be sent to the mobile in the direction specified. |
| Query | The query form of this command is not available. |
| Example | :CONF:CDMA:BS:TCH:PCON:ONCE UP,10 Commands the 4400 to send ten power control bits for the uplink. |

:CONFigure:CDMA:BS:TCH:FCH:TCODE

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:FCH:TCODE <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 2, the maximum value 63. The default value is 8. |
| Description | This command sets the traffic code for the forward link channel. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:BS:TCH:FCH:TCODE 8 :CONF:CDMA:BS:TCH:FCH:TCODE? Value returned: 8. |

:CONFigure:CDMA:BS:TCH:FCH:VMODE

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:TCH:FCH:VMODE <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: LOOPback GENSpeech GENHz400 GENKHz GENAudioSwp SILEnt. Default is LOOPback. |
| Description | Sets the voice mode default configuration. |
| Query | The query form of this command returns the current setting. |
| Example | :CONF:CDMA:BS:TCH:FCH:VMOD GENHz400 :CONF:CDMA:BS:TCH:VMOD? Value returned: GENHz400. |

:CONFigure:CDMA:BS:TCH:FCH:DRATE

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:TCH:FCH:DRATE <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: FULL HALF QUAR-ter EIGHth. Default is FULL. |
| Description | Sets the data rate used on the forward link channel. |
| Query | The query form of this command returns the current setting. |

| | |
|----------------|--|
| Example | :CONF:CDMA:BS:TCH:FCH:DRATe FULL :CONF:CDMA:BS:TCH:DRAT? Value returned: FULL. |
|----------------|--|

:CONFigure:CDMA:BS:TCH:FCH:VOICe:DELaY

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:FCH:VOICe:DELaY <Int1> |
| Parameters | Int1 is an integer number. The parameter's minimum value is 0, the maximum value is 5. The default value is 1. |
| Description | Sets the voice delay used on the forward link channel, in seconds. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:BS:TCH:FCH:VOIC:DEL 2 :CONF:CDMA:BS:TCH:FCH:VOIC:DEL? Value returned: 2. |

:CONFigure:CDMA:BS:TCH:SCH:[FORWARD]:ENCoding

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:SCH:[FORWARD]:ENCoding <Pre-defExp> |
| Parameters | PredefExp is one of the following predefined expressions: CONVen-tional TURBo. Default is CONVen-tional. |
| Description | Sets the forward encoding used on the supplemental channel. |
| Query | The query form of this command returns one of the predefined expressions CONVen-tional TURBo. |
| Example | :CONF:CDMA:BS:TCH:SCH:FORW:ENC TURBo :CONF:CDMA:BS:TCH:SCH:FORW:ENC? Value returned: TURBo. |

:CONFigure:CDMA:BS:TCH:SCH:FORW:DRATe3

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:SCH:FORW:DRATe3 <PredefExp> :CONFigure:CDMA:BS:TCH:SCH:FORW:DRATe4 <PredefExp> :CONFigure:CDMA:BS:TCH:SCH:FORW:DRATe5 <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: FULL DRATe_x2 DRATe_x4 DRATe_x8 DRATe_x16 DRATe_x32. Default is FULL. |
| Description | Sets the data rate used on the supplemental channel for a particular radio configura-tion. |
| Query | The query form of this command returns the current setting. |
| Example | :CONF:CDMA:BS:TCH:SCH:FORW:DRATe3 FULL :CONF:CDMA:BS:TCH:SCH:FORW:DRAT3? Value returned: FULL. :CONF:CDMA:BS:TCH:SCH:FORW:DRATe4 FULL :CONF:CDMA:BS:TCH:SCH:FORW:DRAT4? Value returned: FULL. |

:CONFigure:CDMA:BS:TCH:SCH:FORW:PATtern

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:SCH:FORW:PATtern <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: FIXed PBRs. Default is FIXed. |
| Description | Sets the pattern used on supplemental channel. |
| Query | The query form of this command returns the current setting. |
| Example | :CONF:CDMA:BS:TCH:SCH:FORW:PATtern PBRs :CONF:CDMA:BS:TCH:SCH:FORW:PATT? Value returned: PBRs. |

:CONFigure:CDMA:BS:TCH:SCH:FORW:PATtern:TYPE

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:TCH:SCH:FORW:PATtern:TYPE <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 255. The default value is 255. |
| Description | Sets the type of data pattern used on the supplemental channel. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:BS:TCH:SCH:FORW:PATT:TYPE 255 :CONF:CDMA:BS:TCH:SCH:FORW:PATT:TYPE? Value returned: 255. |

:CONFigure:CDMA:BS:TCH:SCH:REVerse:ENCoding

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:SCH:REVerse:ENCoding <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: CONVen-tional TURBo. Default is CONVen-tional. |
| Description | Sets the reverse encoding used on the supplemental channel. |
| Query | The query form of this command returns the current setting. |
| Example | :CONF:CDMA:BS:TCH:SCH:REV:ENCoding TURBo :CONF:CDMA:BS:TCH:SCH:REV:ENC? Value returned: TURBo. |

:CONFigure:CDMA:BS:TCH:SCH:REV:DRATe3

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:TCH:SCH:REV:DRATe3 <PredefExp> :CONFigure:CDMA:BS:TCH:SCH:REV:DRATe4 <PredefExp> :CONFigure:CDMA:BS:TCH:SCH:REV:DRATe5 <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: FULL DRATe_x2 DRATe_x4 DRATe_x8 DRATe_x16 DRATe_x32. Default is FULL. |
| Description | Sets the data rate used on the supplemental channel for a particular radio configuration. |

| | |
|----------------|---|
| Query | The query form of this command returns the current setting. |
| Example | :CONF:CDMA:BS:TCH:SCH:REV:DRAT3 FULL :CONF:CDMA:BS:TCH:SCH:REV:DRAT3? Value returned: FULL. :CONF:CDMA:BS:TCH:SCH:REV:DRATe4 FULL :CONF:CDMA:BS:TCH:SCH:REV:DRAT4? Value returned: FULL. |

:CONFigure:CDMA:BS:TCH:SCH:REVerse:STATE

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:TCH:SCH:REVerse:STATE <Boolean> |
| Parameters | Boolean is a boolean parameter which can take on the values 0 1. The default is 0. |
| Description | This command turns the reverse channel on (1) or off (0). |
| Query | The query form of this command returns the current setting. |
| Example | :CONF:CDMA:BS:TCH:SCH:REV:STAT 0 :CONF:CDMA:BS:TCH:SCH:REV:STAT? Returns 0, meaning that the supplemental reverse channel is switched off. |

:CONFigure:CDMA:BS:BASE:ID

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:BASE:ID <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 65535. The default value is 39. |
| Description | This command sets the registration toggles low bit between 0 and 1. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:BASE:ID 39 :CONF:CDMA:BS:BASE:ID? Value returned in this example: 0. |

:CONFigure:CDMA:BS:BASE:LAT

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:BASE:LAT <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 1. The default value is 0. |
| Description | This command sets the registration toggles low bit between 0 and 1. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:PARAM:REG 1 :CONF:CDMA:BS:PARAM:REG? Value returned in this example: 0. |

:CONFigure:CDMA:BS:BASE:LONG

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:BASE:LONG <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 1. The default value is 0. |
| Description | This command sets the registration toggles low bit between 0 and 1. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:BASE:LONG 1 :CONF:CDMA:BS:BASE:LONG? Value returned in this example: 0. |

:CONFigure:CDMA:BS:SLOT:IND

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:SLOT:IND <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 7. The default value is 1. |
| Description | This command sets the Registration toggles low bit between 0 and 1. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:SLOT:IND 1 :CONF:CDMA:BS:BASE:ID? Value returned in this example: 0. |

:CONFigure:CDMA:BS:REGZ

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:REGZ <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 4095. The default value is 0. |
| Description | This command sets the registration toggles low bit between 0 and 1. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:REGZ 0 :CONF:CDMA:BS:REGZ? Value returned in this example: 0. |

:CONFigure:CDMA:BS:PARM:REG

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:BS:PARM:REG <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 1. The default value is 0. |
| Description | This command sets the registration toggles low bit between 0 and 1. |

| | |
|----------------|--|
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:PARM:REG 1 :CONF:CDMA:BS:PARM:REG? Value returned in this example: 0. |

:CONFigure:CDMA:BS:REG:PRD

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:REG:PRD <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 1. The default value is 0. |
| Description | This command sets the registration toggles low bit between 0 and 1. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:REG:PRD 0 :CONF:CDMA:BS:REG:PRD? Value returned in this example: 0. |

:CONFigure:CDMA:BS:REG:DIST

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:REG:DIST <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 1. The default value is 0. |
| Description | This command sets the registration toggles low bit between 0 and 1. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:REG:DIST 0 :CONF:CDMA:BS:REG:DIST? Value returned in this example: 0. |

:CONFigure:CDMA:BS:PREF:MSID:TYPE

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:PREF:MSID:TYPE <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 3. The default value is 3. |
| Description | This command sets the Preferred MSID type. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:PREF:MSID:TYPE 3 :CONF:CDMA:BS:REG:DIST? Value returned in this example: 3. |

:CONFigure:CDMA:BS:MAX:CAP:SZ

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:MAX:CAP:SZ <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 7. The default value is 7. |
| Description | This command sets the Preferred MSID type. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:MAX:CAP:SZ 7 :CONF:CDMA:BS:REG:DIST? Value returned in this example: 7. |

:CONFigure:CDMA:BS:PROB:PN:RAN

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:PROB:PN:RAN <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 15. The default value is 0. |
| Description | This command sets the Preferred MSID type. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | :CONF:CDMA:BS:PROB:PN:RAN 0 :CONF:CDMA:BS:PROB:PN:RAN? Value returned in this example: 7. |

:CONFigure:CDMA:BS:ACCess:PNOMinal

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:ACCess:PNOMinal <Reall> |
| Parameters | Reall is a floating point real number. The minimum value is -24, the maximum value 7. The minimum resolution is 1. The default value is 0. |
| Description | This command sets the access probe's nominal power of the Willtek 4400 (i.e. simulated base station). The physical dimension is dBm. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:BS:ACCess:PNOMinal 0 :CONF:CDMA:BS:ACC:PNOM? Value returned: 0. |

:CONFigure:CDMA:BS:ACCess:PINit

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:ACCess:PINit <Int1> |
| Parameters | Int1 is an integer number. The minimum value is -16, the maximum value 15. The default value is 0. |
| Description | This command sets the initial power used on the access probe. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |

Example :CONF:CDMA:BS:ACC:PIN 3
:CONF:CDMA:BS:ACC:PIN?
Value returned: 3.

:CONFigure:CDMA:BS:ACCess:PSTep

Syntax :CONFigure:CDMA:BS:ACCess:PSTep <Int1>
Parameters Int1 is an integer number. The minimum value is 0, the maximum value 7. The default value is 0.
Description This command sets the power step used on the access probe.
Query The query form of this command will return the current setting of the Willtek 4400.
Example :CONF:CDMA:BS:ACC:PST 0
:CONF:CDMA:BS:ACC:PST?
Value returned: 0.

:CONFigure:CDMA:BS:ACCess:STEPs

Syntax :CONFigure:CDMA:BS:ACCess:STEPs <Int1>
Parameters Int1 is an integer number. The minimum value is 1, the maximum value 16. The default value is 4.
Description This command sets the number of probes to be used with an access probe.
Query The query form of this command will return the current setting of the Willtek 4400.
Example :CONF:CDMA:BS:ACC:STEP 5
:CONF:CDMA:BS:ACC:STEP?
Value returned: 5.

:CONFigure:CDMA:BS:ACCess:RESPonse

Syntax :CONFigure:CDMA:BS:ACCess:RESPonse <Int1>
Parameters Int1 is an integer number. The minimum value is 1, the maximum value 15. The default value is 15.
Description This command sets the number of response sequences.
Query The query form of this command will return the current setting of the Willtek 4400.
Example :CONF:CDMA:BS:ACC:RESP 15
:CONF:CDMA:BS:ACC:RESP?
Value returned: 15.

:CONFigure:CDMA:BS:ACCess:Request

Syntax :CONFigure:CDMA:BS:ACCess:Request <Int1>
Parameters Int1 is an integer number. The minimum value is 1, the maximum value 15. The default value is 15.

| | |
|--------------------|---|
| Description | This command sets the number of request sequences. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:BS:ACC:REQ 15 :CONF:CDMA:BS:ACC:REQ? Value returned: 15. |

:CONFigure:CDMA:BS:ACCess:PREamble

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:ACCess:PREamble <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 1, the maximum value 16. The default value is 1. |
| Description | This command sets the preamble length. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:BS:ACC:PRE 1 :CONF:CDMA:BS:ACC:PRE? Value returned: 1. |

:CONFigure:CDMA:BS:ACCess:TIMEout

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:BS:ACCess:TIMEout <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is 0.16, the maximum value 1.36. The minimum resolution is 0.08. The default value is 1.36. |
| Description | This command sets the access probe's nominal power of the Willtek 4400 (i.e. simulated base station). The physical dimension is microseconds. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:BS:ACCess:TIMEout 0.16 :CONF:CDMA:BS:ACC:TIM? Value returned: 0.16. |

:CONFigure:CDMA:MEASure:GROup:[RFTX]

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:MEASure:GROup:[RFTX] <PredefExp1>[, <PredefExp2>][, ... <PredefExp20>] |
| Parameters | At least one and up to 19 parameters can be given, each with the same range of pre-defined expressions: NONE PPHASerr PHASeerr PEAKerr RMSErr FREQuency POWer PMAGerr MAGerr TIMEerr WQUALity CARRierfeed IQIMbalance CDPOwer CDPHase CDTime CDIPower CDINoise CDQPowEr CDQNoise. Default is NONE in all cases. |
| Description | Sets the group of measurements to be carried out. Up to 19 measurements can be grouped. |
| Query | The query form of this command returns the measurements that have been grouped. |

| | |
|----------------|--|
| Example | <pre>:CONF:CDMA:MEAS:GRO:RFTX WQUA, POWer :CONF:CDMA:MEAS:GRO? Returned string: WQUA, POW.</pre> |
|----------------|--|

:CONFigure:CDMA:MEAS:RFRX:FER[MAXimum]

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:MEAS:RFRX:FER[MAXimum] <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 10, the maximum value 10000. The default value is 1000. |
| Description | This command sets the maximum number of frames to use in the frame error rate test. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | <pre>:CONF:CDMA:MEAS:RFRX:FER:MAX 10000 :CONF:CDMA:MEAS:RFRX:FER:MAX? Value returned: 10000.</pre> |

:CONFigure:CDMA:MEAS:RFRX:FER:CONFidence

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:MEAS:RFRX:FER:CONFidence <Reall> |
| Parameters | Reall is a floating point real number. The minimum value is 90, the maximum value 100. The minimum resolution is 0.5. The default value is 95.0. |
| Description | This command sets the percentage used to indicate if FER test passed with a certain confidence level. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | <pre>:CONF:CDMA:MEAS:RFRX:FER:CONFidence 95 :CONF:CDMA:MEAS:RFRX:FER:CONF? Value returned: 95.</pre> |

:CONFigure:CDMA:MEAS:RFRX:FER:LIMit:[MAXimum]

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:MEAS:RFRX:FER:LIMit:[MAXimum] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value is 0, the maximum value 5. The minimum resolution is 0.1. The default value is 0.5. |
| Description | This command sets the percentage used when the 4400 indicates if the FER test is failed. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | <pre>:CONF:CDMA:MEAS:RFRX:FER:LIM:MAX 5.0 :CONF:CDMA:MEAS:RFRX:FER:LIM? Value returned: 5.0.</pre> |

:CONFigure:CDMA:MEAS:RFRX:FER:ENABLE

| | |
|---------------|--|
| Syntax | :CONFigure:CDMA:MEAS:RFRX:FER:ENABLE <PredefExp> |
|---------------|--|

| | |
|--------------------|---|
| Parameters | <code>PredefExp</code> is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | Enables or disables the limit that is applied to the FER test. |
| Query | The query form of this command returns the current setting. |
| Example | <code>:CONF:CDMA:MEAS:RFRX:FER:ENAB OFF</code> <code>:CONF:CDMA:MEAS:RFRX:FER:ENAB?</code> Value returned: OFF. |

:CONFigure:CDMA:MEAS:RFRX:RFER:PERE

| | |
|--------------------|---|
| Syntax | <code>:CONFigure:CDMA:MEAS:RFRX:RFER:PERE <PredefExp></code> |
| Parameters | <code>PredefExp</code> is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | Sets the reported FER periodic enable flag. |
| Query | The query form of this command returns the current setting. |
| Example | <code>:CONF:CDMA:MEAS:RFRX:RFER:PERE OFF</code> <code>:CONF:CDMA:MEAS:RFRX:RFER:PERE?</code> Value returned: OFF. |

:CONFigure:CDMA:MEAS:RFRX:RFER:THRE

| | |
|--------------------|---|
| Syntax | <code>:CONFigure:CDMA:MEAS:RFRX:RFER:THRE <PredefExp></code> |
| Parameters | <code>PredefExp</code> is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | Sets the reported FER threshold enable flag. |
| Query | The query form of this command returns the current setting. |
| Example | <code>:CONF:CDMA:MEAS:RFRX:RFER:THRE OFF</code> <code>:CONF:CDMA:MEAS:RFRX:RFER:THRE?</code> Value returned: OFF. |

:CONFigure:CDMA:MEAS:RFRX:RFER:THRS

| | |
|--------------------|---|
| Syntax | <code>:CONFigure:CDMA:MEAS:RFRX:RFER:THRS <Int1></code> |
| Parameters | <code>Int1</code> is an integer number. The minimum value is 0, the maximum value 31. The default value is 0. |
| Description | This command sets the threshold used with reported frame error rate. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | <code>:CONF:CDMA:MEAS:RFRX:RFER:THRS 0</code> <code>:CONF:CDMA:MEAS:RFRX:RFER:THRS?</code> Value returned: 0. |

:CONFigure:CDMA:MEAS:RFRX:RFER:REPPeriod

| | |
|--------------------|---|
| Syntax | :CONFigure:CDMA:MEAS:RFRX:RFER:REPPeriod <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 16. The default value is 10. |
| Description | This command sets the report period used with the reported frame error rate. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:MEAS:RFRX:RFER:REPP 12 :CONF:CDMA:MEAS:RFRX:RFER:REPP? Value returned: 12. |

:CONFigure:CDMA:MEAS:RFRX:RFER:DELAy

| | |
|--------------------|--|
| Syntax | :CONFigure:CDMA:MEAS:RFRX:RFER:DELAy <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 31. The default value is 0. |
| Description | This command sets the delay used with the reported frame error rate. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CONF:CDMA:MEAS:RFRX:RFER:DELAy 0 :CONF:CDMA:MEAS:RFRX:RFER:DELAy? Value returned: 0. |

:CONFigure:CDMA:COUPlOSS:STATe

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

:CONFigure:CDMA:COUPlOSS:NAME

| | |
|-------------------|--|
| Syntax | :CONFigure:CDMA: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 parameter. Please note that the data contained in the file need to be activated (using the <code>CONF:COUP:STAT ON</code> command described above) before the data contained in the file specified will have any effect on the measurement results. |
| 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 | <code>:CONFigure:COUPloss:NAME "m7389.cpl"</code> <code>:CONF:COUP:NAME?</code> Value returned: "m7389.cpl" |

:CONFigure:CDMA:COUPloss:INFormation?

| | |
|--------------------|---|
| Syntax | <code>:CONFigure:CDMA:COUPloss:INFormation?</code> |
| 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 | <code>:CONFigure:CDMA:COUPloss:INFormation?</code> Value returned: "Motorola P7389 with Antenna Coupler" |

:CONFigure:CDMA:COUPloss:DATA

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

| | |
|--------------------|---|
| Parameters | <p><code>string1</code> is a comment line related to the coupling loss data. The maximum length of <code>string1</code> is 255 characters.</p> <p><code>realfx</code> are floating point real numbers, giving frequencies in MHz while <code>realax</code> are floating point real numbers, giving the corresponding coupling loss in dB. All <code>realfx</code> 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.</p> <p>The minimum resolution for all <code>realfx</code> values is 10 Hz (0.00001 MHz). The default value for all <code>realfx</code> is 800.0 MHz. The minimum value for all <code>realax</code> is -5.0 dB. The maximum value for all <code>realax</code> is 40.0 dB. The minimum resolution for all <code>realax</code> is 0.01 dB. The default value for all <code>realax</code> is 0.0.</p> <p>Notes</p> <ul style="list-style-type: none"> - 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 <code>realax</code> 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 | <code>:CONFigure:CDMA:COUPlOSS:DATA "Motorola 7389 with cable #23",825.0,15.0,1750.0,19.0</code> |

:CONFigure:AMPS:BS:MAC

| | |
|--------------------|--|
| Syntax | <code>:CONFigure:AMPS:BS:MAC <Real1></code> |
| Parameters | <code>Real1</code> is a floating point real number. The minimum value is 0, the maximum value is 7. The minimum resolution is 1. The default value is 2. |
| Description | This command sets the MAC power level of the mobile under test. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one real number. |
| Example | <code>:CONF:AMPS:BS:MAC 2</code> <code>:CONF:AMPS:BS:MAC?</code> Value returned in this example: 2. |

:CONFigure:AMPS:BS:SAT

| | |
|--------------------|--|
| Syntax | <code>:CONFigure:AMPS:BS:SAT <Real1></code> |
| Parameters | <code>Real1</code> is a floating point real number. The minimum value is 0, the maximum value is 2. The minimum resolution is 1. The default value is 0. |
| Description | This command sets the SAT frequency to be utilized when a voice channel is established. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one real number. |

| | |
|----------------|--|
| Example | <pre>:CONF:AMPS:BS:SAT 2 :CONF:AMPS:BS:SAT? Value returned in this example: 2.</pre> |
|----------------|--|

:CONFigure:AMPS:BS:VCH:ARFCn

| | |
|--------------------|--|
| Syntax | <code>:CONFigure:AMPS:BS:VCH:ARFCn <Int1></code> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 2047. The default value is 29. |
| Description | This command sets the channel number for the voice channel. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one integer number. |
| Example | <pre>:CONF:AMPS:BS:VCH:ARFC 0 :CONF:AMPS:BS:VCH:ARFC? Value returned in this example: 0.</pre> |

CONFigure:AMPS:MEAS:RFTX:FM:DETECTOR

| | |
|--------------------|---|
| Syntax | <code>:CONFigure:AMPS:MEAS:RFTX:FM:DETECTOR <PredefExp1></code> |
| Parameters | PredefExp is one of the following predefined expressions: PEAKpeak PPOSitive PNEGative NONE. Default is PEAKpeak. |
| Description | This command selects the detector type used for deviation 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 | <pre>:CONFigure:AMPS:MEAS:RFTX:FM:DET PPOSitive :CONFigure:AMPS:MEAS:RFTX:FM:DET? Value returned in this example "PPOSitive".</pre> |

:CONFigure:AMPS:MEAS:RFTX:FM:FILTEr

| | |
|--------------------|---|
| Syntax | <code>:CONFigure:AMPS:MEAS:RFTX:FM:FILTEr <PredefExp1></code> |
| Parameters | PredefExp is one of the following predefined expressions: NORMAl CMESsage. Default is NORMAl. |
| Description | This command selects the filter type used for AMPS deviation 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 | <pre>:CONFigure:AMPS:MEAS:RFTX:FM:FILT CMESsage :CONFigure:AMPS:MEAS:RFTX:FM:FILT? Value returned in this example CMES.</pre> |

:CONFigure:AMPS:MEASure:GROup:[RFTX]

| | |
|---------------|--|
| Syntax | <code>CONFigure:AMPS:MEASure:GROup:[RFTX] <PredefExp1>[, <PredefExp2>][, ... <PredefExp20>]</code> |
|---------------|--|

| | |
|--------------------|--|
| Parameters | At least one and up to 5 parameters can be given, each with the same range of pre-defined expressions: NONE SATDev SATFreq STDev STFReq AUD-Dev FREQUENCY POWER. Default is NONE in all cases. |
| Description | Sets the group of measurements to be carried out. Up to 19 measurements can be grouped. |
| Query | The query form of this command returns the measurements that have been grouped. |
| Example | :CONF:AMPS:MEAS:GRO:RFTX SATD, SATFreq, Power :CONF:AMPS:MEAS:GRO? Returned string in this example: SATD, SATF, POW. |

:CONFigure:AMPS:MEASure:GROup:RFRX

| | |
|--------------------|--|
| Syntax | CONFigure:AMPS:MEASure:GROup:RFRX <PredefExp1> [, <PredefExp2>] [, ... <PredefExp20>] |
| Parameters | At least one and up to 2 parameters can be given, each with the same range of pre-defined expressions: NONE SINad ACLevel. Default is NONE in all cases. |
| Description | Sets the group of measurements to be carried out. Up to 19 measurements can be grouped. |
| Query | The query form of this command returns the measurements that have been grouped. |
| Example | :CONF:AMPS:MEAS:GRO:RFRX SINad, ACLevel :CONF:AMPS:MEAS:GRO? Returned string in this example: SIN, ACL. |

CALL subsystem

This subsystem contains commands

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

:CALL:CDMA:BSORiginate

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:BSORiginate |
| Parameters | There are no parameters. |
| Description | Initiates a call setup by the base station (Willtek 4400). |
| Query | There is no query form of this command available. |
| Example | :CALL:CDMA:BSOR This command will initiate a BS-originated call in thte 4400 (i.e. the mobile phone will be called). |

:CALL:CDMA:ABORT

| | |
|---------------|------------------|
| Syntax | :CALL:CDMA:ABORT |
|---------------|------------------|

| | |
|--------------------|--|
| Parameters | There are no parameters. |
| Description | Terminates a call setup attempt by the base station (Willtek 4400). This will also stop the Access Probe measurement (if active) and tear down a call (if on a traffic channel). This is used to halt a pending BS origination. The BS release should be used instead if the BS origination has completed, since it signals the mobile to release prior to tearing down the traffic channel. |
| Query | There is no query form of this command available. |
| Example | :CALL:CDMA:ABOR This command will terminate a BS-originated call in the 4400 (i.e. the mobile and 4400 will be on the Control Channel upon completion). |

:CALL:CDMA:BSRelease

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:BSRelease |
| Parameters | There are no parameters. |
| Description | This command will make the Willtek 4400 terminate the ongoing call. |
| Query | There is no query form of this command available. |
| Example | :CALL:CDMA:BSR This command will release the call. |

:CALL:CDMA:REGistration

| | |
|--------------------|--|
| Syntax | :CALL:CDMA:REGistration |
| Parameters | There are no parameters. |
| Description | This command causes the mobile to try to register with the Willtek 4400. |
| Query | There is no query form of this command available. |
| Example | :CALL:CDMA:REG |

:CALL:CDMA:REGistration:CLEAr

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:REGistration:CLEAr |
| Parameters | There are no parameters. |
| Description | This command causes the Willtek 4400 to erase any stored information about a previous registration by a mobile. |
| Query | There is no query form of this command available. |
| Example | :CALL:CDMA:REG:CLEAr |

:CALL:CDMA:HANDoff[:IMMediate]

| | |
|---------------|--------------------------------|
| Syntax | :CALL:CDMA:HANDoff[:IMMediate] |
|---------------|--------------------------------|

| | |
|--------------------|--|
| Parameters | There are no parameters. |
| Description | This command causes the Willtek 4400 to send a handoff to the mobile. |
| Query | There is no query form of this command available. |
| Example | :CALL:CDMA:HAND:TCH 25 :CALL:CDMA:HAND:IMM Performs a handoff to channel 25. |

:CALL:CDMA:HANDoff:BAND

| | |
|--------------------|--|
| Syntax | :CALL:CDMA:HANDoff:BAND <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. Default is US800. |
| Description | This command changes the band setting used when a :CALL:HANDoff:IMMEDIATE is issued. |
| Query | There is no query form of this command available. |
| Example | :CALL:CDMA:HANDoff:BAND USPCs |

:CALL:CDMA:HANDoff:TCH

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:HANDoff:TCH <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 2047. The initial default is 29, which changes to the current channel during a call setup. |
| Description | This command sets the channel number for the traffic channel used during a handoff. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CALL:CDMA:HAND:TCH 30 :CALL:CDMA:HAND:TCH? Value returned: 30. |

:CALL:CDMA:HANDoff:RCONfig

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:HANDoff:RCONfig <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 1, the maximum value 5. The initial default is 1, which changes to the current radio configuration during a call setup. |
| Description | This command sets the forward radio configuration used in a handoff. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CALL:CDMA:HAND:RCON 3 :CALL:CDMA:HAND:RCON? Value returned: 3. |

:CALL:CDMA:HANdoff:SOPTion

| | |
|--------------------|--|
| Syntax | :CALL:CDMA:HANdoff:SOPTion <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 1, the maximum value 32768. The default value is initialized to the current value set by :CONF:CDMA:BS:TCH:SOPT. |
| Description | This command sets the service option to be used in a service option handoff. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CALL:CDMA:HAND:SOPT 3 :CALL:CDMA:HAND:SOPT? Value returned: 3. |

:CALL:CDMA:MSINfo:ALL

| | |
|--------------------|--|
| Syntax | :CALL:CDMA:MSINfo:ALL |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This command reads out all MS Info result values with one query. The query will return a string containing 11 values. 1. NUMBer, string 2. MIN IMSI, string 3. ESN, string 4. SCM, integer 5. Protocol Version, integer 6. Slot Cycle Index, integer 7. TCALs, BOOL 8. RTYPE, integer 9. Manufacture Code, integer 10. Serial Number, integer 11. MSCClass, integer |
| Example | :CALL:CDMA:MSINfo:ALL? The string delivered back is for example: "5551234","001010123456789","B04F575F",106,3, 1,1,0,82,166398,3". |

:CALL:CDMA:MSINfo:NUMBer

| | |
|--------------------|--|
| Syntax | :CALL:CDMA: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:CDMA:MSINfo:NUMB? The string delivered back is, for example: "5551234". |

:CALL:CDMA:MSInfo:IMSI

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

:CALL:CDMA:MSInfo:MSClass

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:MSInfo:MSClass |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back the code representing the mobile's power class. The query will deliver back one integer. |
| Example | :CALL:CDMA:MSInfo:MSClass? The integer delivered back could be: 3. |

:CALL:CDMA:MSInfo:ESN:[IMMEDIATE]

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:MSInfo:ESN:[IMMEDIATE] |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back a string containing the mobile's electronic serial number. |
| Example | :CALL:CDMA:MSInfo:ESN:IMM? The string delivered back is: "B04F575F". |

:CALL:CDMA:MSInfo:ESN:MCode

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:MSInfo:ESN:MCode |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back the mobile's manufacturer identity code. The query will deliver back an integer. |
| Example | :CALL:CDMA:MSInfo:ESN:MCode? The integer delivered back is: 82. |

:CALL:CDMA:MSInfo:ESN:SNUMber

| | |
|--------------------|--|
| Syntax | :CALL:CDMA:MSInfo:ESN:SNUMber |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back the mobile's equipment serial number. The query will deliver back an integer. |
| Example | :CALL:CDMA:MSInfo:ESN:SNUM? The integer delivered back is: 428542. |

:CALL:CDMA:MSInfo:SCM

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:MSInfo:SCM |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back the mobile's station class mark. The query will deliver back an integer. |
| Example | :CALL:CDMA:MSInfo:ESN:SNUM? The integer delivered back is: 106. |

:CALL:CDMA:MSInfo:PREVision

| | |
|--------------------|--|
| Syntax | :CALL:CDMA:MSInfo:PREVision |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back the mobile's protocol revision. The query will deliver back an integer. |
| Example | :CALL:CDMA:MSInfo:PREV? The integer delivered back is: 3. |

:CALL:CDMA:MSInfo:SCIndex

| | |
|--------------------|--|
| Syntax | :CALL:CDMA:MSInfo:SCIndex |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back the slot cycle index. The query will deliver back an integer. |
| Example | :CALL:CDMA:MSInfo:SCIN? The integer delivered back is: 1. |

:CALL:CDMA:MSInfo:TCALs

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:MSInfo:TCALs |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back a boolean indicating if calls are terminated. The query will deliver back a boolean. |
| Example | :CALL:CDMA:MSInfo:TCAL? The boolean delivered back is: 1. |

:CALL:CDMA:MSInfo:RTYPE

| | |
|--------------------|---|
| Syntax | :CALL:CDMA:MSInfo:RTYPE |
| Parameters | There are no parameters. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back the registration type. The query will deliver back an integer. |
| Example | :CALL:CDMA:MSInfo:RTYPE? The integer delivered back is: 0. |

:CALL:CDMA:MSInfo:CAPable?

| | |
|--------------------|--|
| Syntax | :CALL:CDMA:MSInfo:CAPable? <PredefExp> |
| Parameters | PredefExp is one of US800 USPCS TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. |
| Description | There is only a query form of this command available. |
| Query | This query will deliver back one of the following expressions: YES NO UNKNown. It reflects the mobile's capability to set up calls in the respective frequency band. |
| Example | :CALL:CDMA:MSInfo:CAP? USPCS The string delivered back is: YES. |

:CALL:AMPS:HANdoff:[IMMediate]

| | |
|--------------------|--|
| Syntax | :CALL:AMPS:HANdoff:[IMMediate] |
| Parameters | There are no parameters. |
| Description | This command causes the Willtek 4400 to send a handoff to the mobile. |
| Query | There is no query form of this command available. |
| Example | :CALL:AMPS:HAND:VCH 25 :CALL:AMPS:HAND:IMM Performs a handoff to channel 25. |

:CALL:AMPS:HANDoff:VCH

| | |
|--------------------|---|
| Syntax | :CALL:AMPS:HANDoff:VCH <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 1023. The initial default is 29, which changes to the current channel during a call setup. |
| Description | This command sets the channel number for the voice channel used during a handoff. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CALL:AMPS:HAND:VCH 30 :CALL:AMPS:HAND:VCH? Value returned in this example: 30. |

:CALL:AMPS:HANDoff:SAT

| | |
|--------------------|---|
| Syntax | :CALL:AMPS:HANDoff:SAT <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 2. The initial default is 0, which changes to the current radio configuration during a call setup. |
| Description | This command sets the SAT frequency used in a handoff. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CALL:AMPS:HAND:SAT 2 :CALL:AMPS:HAND:SAT? Value returned in this example: 2. |

:CALL:AMPS:HANDoff:MAC

| | |
|--------------------|--|
| Syntax | :CALL:AMPS:HANDoff:MAC <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 10. The default value is initialized to the current value set by :CONF:AMPS:BS:MAC. |
| Description | This command sets the MAC level of the mobile during a handoff. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :CALL:AMPS:HAND:MAC :CALL:AMPS:HAND:MAC? Value returned in this example: 3. |

:CALL:AMPS:SAT:REP:STATe

| | |
|--------------------|--|
| Syntax | :CALL:AMPS:SAT:REP:STATe <PredefExp> |
| Parameters | PredefExp can take on one of the following values: OFF ON. The default is OFF. |
| Description | This command switches receive SAT Status reports on. |
| Query | This command enables/disables the receive SAT Status reports. |
| Example | :CALL:AMPS:SAT:REP:STAT ON Switches the SAT Status reports on. |

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.

:MEASure[:CONT]:AFANalyser:GROup

| | |
|--------------------|--|
| Syntax | :MEASure[:CONT]:AFANalyser:GROup |
| Parameters | There are no parameters. |
| Description | <p>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.</p> <p>Notes:</p> <ul style="list-style-type: none"> - 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 287 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 294. |
| Query | <p>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).</p> <p>The single measurement result values are separated by commas.</p> |
| Example | <p>:CONF:MEAS:GRO:AFAN SIN,FREQ :MEASure:CONTinuous:AFANalyser:GROup?</p> <p>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".</p> <p>Because of the internal order, the first measurement result value delivered back is the audio frequency, the second one the SINAD.</p> |

:MEASure[:CONT]:AFANalyser:ALL

| | |
|-------------------|--------------------------------|
| Syntax | :MEASure[:CONT]:AFANalyser:ALL |
| Parameters | There are no parameters. |

| | |
|--------------------|--|
| Description | <p>Starts a continuous measurement of the most important audio tests. To read out the latest measurement results, use the <code>:FETCh:AFAN:ALL</code> command.</p> <p>The audio tests performed by this command are:</p> <p>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 square 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.</p> <p>Notes:</p> <ul style="list-style-type: none"> – 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 287 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 294. |
| Query | <p>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:</p> <ol style="list-style-type: none"> 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 | <p>:MEASure:CONTinuous:AFANalyser:ALL?</p> <p>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:</p> <p>"0.7,0.25,0.0,1000.0,0.3,50.5".</p> |

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

| | |
|--------------------|--|
| Syntax | <code>:MEAS[:CONT]:AFAN:ACVoltage:PEAKp</code> |
| Parameters | There are no parameters. |
| Description | <p>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 <code>:FETCh:AFAN:ACV:PEAK</code> command.</p> <p>Note: Any AF test command needs the Audio Option to be installed on your Willtek 4400.</p> |
| Query | <p>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}).</p> |
| Example | <p>:MEAS:CONT:AFAN:ACV:PEAKp?</p> <p>String delivered back: "0.7".</p> |

: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:SIAd

| | |
|--------------------|--|
| Syntax | :MEASure[:CONT]:AFANalyser:SIAd |
| 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:SIAd 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:SIAd? 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 <ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> - 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: <ul style="list-style-type: none"> - 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 <code>:FETCh:PSUP:APOW?</code> 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 | <code>:MEAS:PSUP:APOW?</code> 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 | <code>:MEASure[:CONTInuous]:PSUPply:ACURrent</code> |
| 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 <code>:FETCh:PSUP:ACUR</code> 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 | <code>:MEAS:PSUP:ACUR?</code> 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 | <code>:MEASure[:CONTInuous]:PSUPply:PCURrent</code> |
| 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 <code>:FETCh:PSUP:PCUR</code> 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 | <code>:MEAS:PSUP:PCUR?</code> 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: <ul style="list-style-type: none"> - 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 287 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 294. - More information regarding ARRay measurements can be found in section "MEASure:ARRay" on page 290. |
| 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 | <p>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 <code>:FETCh:AFAN:ALL</code> command. The tests performed by this command are:</p> <p>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 square 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.</p> <p>Notes:</p> <ul style="list-style-type: none"> - 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 287 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 294. |
| Query | <p>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:</p> <ol style="list-style-type: none"> 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 | <p>:MEASure:ARRay:AFANalyser:ALL? 2</p> <p>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:</p> <p>"0.7,0.25,0.0,1000.0,0.3,50.5,0.75,0.3,0.0,1000.0,0.25,50.9"</p> |

:MEAS:ARRay:AFAN:ACVoltage:PEAKp

| | |
|--------------------|--|
| Syntax | <code>:MEAS:ARRay:AFAN:ACVoltage:PEAKp <int1></code> |
| Parameters | <p>int1 is an integer. The minimum value for int1 is 0, the maximum value is 100. The default value for int1 is 0.</p> |
| Description | <p>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 <code>:FETCh:AFAN:ACV:PEAK</code> command.</p> <p>Note: Any AF test command will need the Audio Option to be installed on your Willtek 4400.</p> |

| | |
|----------------|---|
| Query | The query form of this command will perform the measurement the specified number of times (int1 parameter). As soon as all measurements have been completed, all measurement result values will be returned in a string. The string delivered back will contain int1 floating point real numbers. The physical dimension of the measurement result values is volt, measured peak-to-peak (V_{pp}). The single measurement results are separated by commas. |
| Example | :MEAS:ARR:AFANalyser: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:AFANalyser: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:AFANalyser:FREQ? 5 The string returned is: "1000.0,1000.5,1000.9,999.9,1000.0". |

:MEASure:ARRay:AFAN:DISToRtion

| | |
|--------------------|--|
| Syntax | <code>:MEASure:ARRay:AFAN:DISToRtion <int1></code> |
| Parameters | <code>int1</code> is an integer. The minimum value for <code>int1</code> is 0, the maximum value is 100. The default value for <code>int1</code> 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 <code>int1</code> parameter). To read out the entire measurement results array, use the <code>:FETCh:AFAN:DIST</code> 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 (<code>int1</code> 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 <code>int1</code> floating point real numbers. The physical dimension of the measurement result values is percentage (%). The single measurement results are separated by commas. |
| Example | <code>:MEASure:ARRay:AFANalyser:DIST? 5</code> The string returned is: "0.5,0.7,1.1,1.2,0.9". |

:MEASure:ARRay:AFANalyser:SINad

| | |
|--------------------|--|
| Syntax | <code>:MEASure:ARRay:AFANalyser:SINad <int1></code> |
| Parameters | <code>int1</code> is an integer. The minimum value for <code>int1</code> is 0, the maximum value is 100. The default value for <code>int1</code> is 0. |
| Description | Performs the measurement of the SINAD (i.e. signal to noise ratio) a specific number of times (set with the <code>int1</code> parameter). To read out the entire measurement results array, use the <code>:FETCh:AFAN:SIN</code> 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 (<code>int1</code> 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 <code>int1</code> floating point real numbers. The physical dimension of the measurement result values is dB. The single measurement results are separated by commas. |
| Example | <code>:MEASure:ARRay:AFANalyser:SINad? 5</code> The string returned is: "50.5,50.0,49.1,51.2,50.9". |

:MEASure:ARRay:PSUPply:GRoup

| | |
|-------------------|---|
| Syntax | <code>:MEASure:ARRay:PSUPply:GRoup <int1></code> |
| Parameters | <code>int1</code> is an integer. The minimum value for <code>int1</code> is 0, the maximum value is 100. The default value for <code>int1</code> is 0. |

| | |
|--------------------|---|
| Description | <p>Performs the power supply measurements specified with the <code>:CONF:MEAS:GRO:PSUP</code> command for a specific number of times (set with the int1 parameter). To read out the entire measurement results array, use the <code>:FETCh:PSUP:GROup</code> command.</p> <p>Notes</p> <ul style="list-style-type: none"> - 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 294. |
| Query | <p>The query form of this command will start the sequence of power/current consumption measurements as specified with the <code>:CONF:MEAS:GRO:PSUP</code> 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 <code>:MEAS:PSUP:ALL</code>).</p> |
| Example | <p><code>:CONF:MEAS:GRO:PSUP ACUR,APOW</code> <code>:MEASure:ARRay:PSUP:GROup?</code></p> <p>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.</p> |

:MEASure:ARRay:PSUPply:ALL

| | |
|--------------------|---|
| Syntax | <code>:MEASure:ARRay:PSUPply:ALL <int1></code> |
| Parameters | <p>int1 is an integer. The minimum value for int1 is 0, the maximum value is 100. The default value for int1 is 0.</p> |
| Description | <p>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 <code>:FETCh:PSUP:ALL</code> command.</p> <p>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.</p> <p>Notes</p> <ul style="list-style-type: none"> - 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 294. |
| Query | <p>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:</p> <ul style="list-style-type: none"> - 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:CDMA[:CONTInuous]:RFTX:GROup

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:GROup |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the RF TX tests. |
| Query | The query form of this command will start the sequence of RF TX measurements. |
| Example | :CONF:CDMA:MEAS:GRO:RFTX WQU, POW :MEASure:CDMA:RFTX:GROup? In this example, the group of measurements is defined by waveform quality and RF power results. |

:MEASure:CDMA[:CONTInuous]:RFTX:FREQuency:ERRor

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:FREQuency:ERRor |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the frequency error. |
| Query | The query form of this command will start the sequence of frequency error measurements. |
| Example | :MEASure:CDMA:RFTX:FREQ:ERR? Starts a frequency error measurement and returns the result. |

:MEASure:CDMA[:CONTInuous]:RFTX:FREQuency:UNITs

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:FREQuency:UNITs <Pre-defExp> |
| Parameters | PredefExp is a predefined expression: HZ KHZ MHZ. Default is HZ. |
| Description | Sets the units used in returning results of the frequency error measurement. |
| Query | The query form of this command will return the current stored settings. |
| Example | :MEASure:CDMA:RFTX:FREQ:UNIT HZ :MEASure:CDMA:RFTX:FREQ:UNITs? Returns the following string: HZ |

:MEASure:CDMA[:CONTInuous]:RFTX:POWer

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:POWer |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the mobile's RF power. |
| Query | The query form of this command will return the float value for power. |
| Example | :MEASure:CDMA:RFTX:POWer? Starts an RF power measurement and returns the result (in dBm), e.g. -45.3. |

:MEASure:CDMA[:CONTInuous]:RFTX:MINimum

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:MINimum |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the minimum RF power. |
| Query | The query form of this command will return the float value. |
| Example | :MEASure:CDMA:RFTX:MINimum? Starts a minimum RF power measurement and returns the result, e.g. -46.1. |

:MEASure:CDMA[:CONTInuous]:RFTX:MAXimum

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:MAXimum |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the maximum RF power. |
| Query | The query form of this command will return the float value. |
| Example | :MEASure:CDMA:RFTX:MAXimum? Starts the maximum RF power measurement and returns the result in a string: "-46.1" |

:MEASure:CDMA[:CONTInuous]:RFTX:STANdby

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:STANdby |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the standby RF power. |
| Query | The query form of this command will return the float value. |
| Example | :MEASure:CDMA:RFTX:STANdby? Starts the standby RF power measurement and returns the result in a string, e.g. "-54.9" |

:MEASure:CDMA[:CONTInuous]:RFTX:ACCess

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:ACCess |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the access channel power. |
| Query | The query form of this command will return the float value. |
| Example | :MEASure:CDMA:RFTX:ACCess? Starts the access channel power measurement and returns a string containing the result, e.g. "-46.1" |

:MEASure:CDMA[:CONTInuous]:RFTX:CODEdomain[:IMMEDIATE]

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:CODEdomain[:IMMEDIATE] |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the code domain power. |
| Query | The query form of this command will return a predefined expression: PASS FAIL NONE. |
| Example | :MEASure:CDMA:RFTX:CODE:IMM? Starts the measurement and returns a string containing the result. |

:MEASure:CDMA[:CONTInuous]:RFTX:CODE:IPOW

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:CODE:IPOW |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the code domain I channel power. |
| Query | The query form of this command will return 16 floating point values. |
| Example | :MEASure:CDMA:RFTX:CODE:IPOW? Starts the measurement and returns a string containing the results (in dB) for the 16 code channels. |

:MEASure:CDMA[:CONTInuous]:RFTX:CODE:INOIse

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:CODE:INOIse |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the code domain I channel noise. |
| Query | The query form of this command will return 16 floating point values. |
| Example | :MEASure:CDMA:RFTX:CODE:INOI? Starts the I channel noise measurement and returns a string containing the results (in dB) for the 16 code channels. |

:MEASure:CDMA[:CONTInuous]:RFTX:CODE:QPOWer

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:CODE:QPOWer |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the code domain Q channel power. |
| Query | The query form of this command will return 16 floating point values. |
| Example | :MEASure:CDMA:RFTX:CODE:QPOW? Starts the measurement and returns a string with the results for the 16 code channels. |

:MEASure:CDMA[:CONTInuous]:RFTX:CODE:QNOIse

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:CODE:QNOIse |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the code domain Q channel noise. |
| Query | The query form of this command will return 16 floating point values for the 16 code channels. |
| Example | :MEASure:CDMA:RFTX:CODE:QNOIse? |

:MEASure:CDMA[:CONTInuous]:RFTX:GATE

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:GATed |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the gated power/time template measurement. |
| Query | The query form of this command will return a predefined expression: PASS FAIL NONE. |
| Example | :MEASure:CDMA:RFTX:GATed:IMM? |

:MEASure:CDMA[:CONTInuous]:RFTX:GATed:AVERAge

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:GATed:AVERAge |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the average power. |
| Query | The query form of this command will return a floating point value. |
| Example | :MEASure:CDMA:RFTX:GATed:AVER? |

:MEASure:CDMA[:CONTInuous]:RFTX:GATed:DROOp

| | |
|---------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:GATed:DROOp |
|---------------|---|

| | |
|--------------------|--|
| Parameters | There are no parameters. |
| Description | Starts a continuous gated power measurement. |
| Query | The query form of this command will return a floating point value. |
| Example | :MEASure:CDMA:RFTX:GATed:DROOp? |

:MEASure:CDMA[:CONTInuous]:RFTX:GATed:PEAk

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:GATed:PEAk |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the gated power measurement. |
| Query | The query form of this command will return a floating point value. |
| Example | :MEASure:CDMA:RFTX:GATed:PEAk? |

:MEASure:CDMA[:CONTInuous]:RFTX:OPEN

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:OPEN |
| Parameters | There are no parameters. |
| Description | Starts a continuous open loop measurement. |
| Query | The query form of this command will return a predefined expression: PASS FAIL NONE. |
| Example | :MEASure:CDMA:RFTX:OPEN:IMM? Returns a pass/fail indication as a result. |

:MEASure:CDMA[:CONTInuous]:RFTX:OPEN:INITial

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:OPEN:INITial |
| Parameters | There are no parameters. |
| Description | Starts a continuous open loop measurement. |
| Query | The query form of this command will return a floating point value. |
| Example | :MEASure:CDMA:RFTX:OPEN:INIT? Returns the intial power. |

:MEASure:CDMA[:CONTInuous]:RFTX:CLOSeD

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:CLOSeD |
| Parameters | There are no parameters. |
| Description | Starts a continuous closed loop measurement. |

| | |
|----------------|--|
| Query | The query form of this command will return a predefined expression: PASS FAIL NONE. |
| Example | :MEASure:CDMA:RFTX:CLOSeD:IMM? |

:MEASure:CDMA[:CONTInuous]:RFTX:CLOSeD:INITial

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:CLOSeD:INITial |
| Parameters | There are no parameters. |
| Description | Starts a continuous closed loop measurement. |
| Query | The query form of this command will return a floating point value for the initial power. |
| Example | :MEASure:CDMA:RFTX:CLOSeD:INIT? |

:MEASure:CDMA[:CONTInuous]:RFTX:CLOSeD:MAXimum

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:CLOSeD:MAXimum |
| Parameters | There are no parameters. |
| Description | Starts a continuous closed loop measurement. |
| Query | The query form of this command will return a floating point value. |
| Example | :MEASure:CDMA:RFTX:CLOSeD:MAXimum? |

:MEASure:CDMA[:CONTInuous]:RFTX:CLOSeD:MINimum

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:CLOSeD:MINimum |
| Parameters | There are no parameters. |
| Description | Starts a continuous closed loop measurement. |
| Query | The query form of this command will return a floating point value. |
| Example | :MEASure:CDMA:RFTX:CLOSeD:MINimum? |

:MEASure:CDMA[:CONTInuous]:RFTX:WQUality

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:WQUality |
| Parameters | There are no parameters. |
| Description | Starts a continuous closed loop measurement. |
| Query | The query form of this command will return a floating point value. |
| Example | :MEASure:CDMA:RFTX:WQUality? Returns the waveform quality, e.g. 0.9989. |

:MEAS:CDMA[:CONTInuous]:RFTX:CDPhase

| | |
|--------------------|--|
| Syntax | :MEAS:CDMA[:CONTInuous]:RFTX:CDPhase? [<PredefExp>] |
| Parameters | The optional PredefExp is one of the predefined expressions ALL RPICH RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. The default is ALL. The channels are: RPICH = R_PICH (W_0^{32}) = pilot channel RDCCh = R_DCCH (W_8^{16}) = dedicated control channel RSCH2x2 = R_SCH2_x2 (W_6^8) = supplementary channel 2 RSCH2x4 = R_SCH2_x4 (W_2^4) = supplementary channel 2 RFCH = R_FCH (W_4^{16}) = fundamental channel RCCCh = R_CCCH (W_2^8) = common control channel RSCH1x4 = R_SCH1_x4 (W_2^4) = supplementary channel 1 RSCH1x8 = R_SCH1_x8 (W_1^2) = supplementary channel 1 |
| Description | Starts a Reverse Pilot to Code Channel Phase Offset measurement. |
| Query | The query form of this command can have an optional parameter specifying which channel to measure. It will return either a floating point value for the selected channel, or eight floating point values for all code channels. The unit for the value is radians. |
| Example | :MEASure:CDMA:RFTX:CDPhase? ALL |

:MEAS:CDMA[:CONTInuous]:RFTX:CDPower

| | |
|--------------------|--|
| Syntax | :MEAS:CDMA[:CONTInuous]:RFTX:CDPower? [<PredefExp>] |
| Parameters | The optional PredefExp is one of the predefined expressions ALL RPICH RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. The default is ALL. |
| Description | The commands starts a Reverse Pilot to Code Channel Power Offset measurement. |
| Query | The query form of this command can have an optional parameter specifying which channel to measure. It will return either one or eight floating point values for the results in dB. |
| Example | :MEASure:CDMA:RFTX:CDPower? ALL |

:MEAS:CDMA[:CONTInuous]:RFTX:CDTime

| | |
|--------------------|--|
| Syntax | :MEAS:CDMA[:CONTInuous]:RFTX:CDTime? [<PredefExp>] |
| Parameters | The optional PredefExp is one of the predefined expressions ALL RPICH RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. The default is ALL. |
| Description | The commands starts a Reverse Pilot to Code Channel Time Offset measurement. |
| Query | The query form of this command can have an optional parameter specifying which channel to measure. It will return either one or eight floating point values expressing the result in microseconds. |
| Example | :MEASure:CDMA:RFTX:CDTime? ALL |

:MEASure:CDMA[:CONTInuous]:RFTX:TIME:OFFSet

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:TIME:OFFSet |
| Parameters | There are no parameters. |
| Description | Starts a continuous time offset measurement. |
| Query | The query form of this command will return a floating point value for the measurement time offset. |
| Example | :MEASure:CDMA:RFTX:Time:OFFSet? Returns a string containing the result, e.g. -4.334. |

:MEASure:CDMA[:CONTInuous]:RFTX:VERRor:RMS

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:VERRor:RMS |
| Parameters | There are no parameters. |
| Description | Starts a continuous RMS vector error measurement. |
| Query | The query form of this command will return a floating point value; the unit is percent. |
| Example | :MEASure:CDMA:RFTX:VERRor:RMS? Returns the measured RMS vector error, e.g. 100.1. |

:MEASure:CDMA[:CONTInuous]:RFTX:VERRor:PEAK

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:VERRor:PEAK |
| Parameters | There are no parameters. |
| Description | Starts a continuous peak vector error measurement. |
| Query | The query form of this command will return a floating point value in percent. |
| Example | :MEASure:CDMA:RFTX:VERRor:PEAK |

:MEASure:CDMA[:CONTInuous]:RFTX:VERRor:OOFfset

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFTX:VERRor:OOFfset |
| Parameters | There are no parameters. |
| Description | Starts a continuous origin offset vector error measurement. |
| Query | The query form of this command will return a floating point value with the result in dBc. |
| Example | :MEASure:CDMA:RFTX:VERRor:OOFfset? Returns a string with the origin offset in dBc, e.g. -40.1. |

:MEASure:CDMA[:CONTinuous]:RFTX:VERRor:IQIMbalance

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTinuous]:RFTX:VERRor:IQIMbalance |
| Parameters | There are no parameters. |
| Description | Starts a continuous IQ Imbalance vector error measurement. |
| Query | The query form of this command will return a floating point value with the result in dB. |
| Example | MEAS:CDMA:ARR:RFTX:IQIM? 20 :CALC:CDMA:RFTX:VERR:IQIM:LIM? Returns 0 if all the 20 results are within limits. |

:MEASure:CDMA[:CONTinuous]:RFTX:VERRor:MAGNitude:RMS

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTinuous]:RFTX:VERRor:MAGNitude:RMS |
| Parameters | There are no parameters. |
| Description | Starts a continuous RMS magnitude vector error measurement. |
| Query | The query form of this command will return a floating point value with the result in percent. |
| Example | :MEASure:CDMA:RFTX:VERRor:MAGNitude:RMS? |

:MEASure:CDMA[:CONTinuous]:RFTX:VERRor:MAGNitude:PEAK

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTinuous]:RFTX:VERRor:MAGNitude:PEAK |
| Parameters | There are no parameters. |
| Description | Starts a continuous peak magnitude vector error measurement. |
| Query | The query form of this command will return a floating point value with the result in percent. |
| Example | :MEASure:CDMA:RFTX:VERRor:MAGNitude:PEAK? |

:MEASure:CDMA[:CONTinuous]:RFTX:VERRor:PHASe:RMS

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTinuous]:RFTX:VERRor:PHASe:RMS |
| Parameters | There are no parameters. |
| Description | Starts a continuous RMS phase vector error measurement. |
| Query | The query form of this command will return a floating point value with the result in degrees. |
| Example | :MEASure:CDMA:RFTX:VERRor:PHASe:RMS? |

:MEASure:CDMA[:CONTinuous]:RFTX:VERRor:PHASe:PEAK

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTinuous]:RFTX:VERRor:PHASe:PEAK |
| Parameters | There are no parameters. |
| Description | Starts a continuous PEAK phase vector error measurement. |
| Query | The query form of this command will return a floating point value with the result in degrees. |
| Example | :MEASure:CDMA:RFTX:VERRor:PHASe:PEAK? |

:MEAS:CDMA[:CONT]:RFSpectrum:ACPM:MODulation

| | |
|--------------------|---|
| Syntax | :MEAS:CDMA[:CONT]:RFSpectrum:ACPM:MODulation |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the ACPM modulation spectrum. |
| Query | The query form of this command starts the measurement and delivers a string, containing 25 measurement result values separated by commas (12 offsets in both directions from the carrier, plus the result at the carrier itself). These measurement result values are floating point real numbers with the physical dimension dBc (dB relative to the carrier, being the transmission frequency of the mobile). |
| Example | :MEAS:CDMA:CONT:RFSP:ACPM:MODulation? |

:MEAS:CDMA[:CONT]:MSP[:CURR]

| | |
|--------------------|---|
| Syntax | :MEAS:CDMA[:CONT]: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: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: |
| 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 | <pre>:MEAS:CDMA:CONT:MSpectrum?</pre> <p>The array returned in this example is: "-62.0,-61.5,-62.2, ...-0.4,0.0,-0.3,...-62.4".</p> |
|----------------|---|

:MEAS:CDMA[:CONT]:MSP:AVG

| | |
|--------------------|---|
| Syntax | <pre>:MEAS:CDMA[:CONT]:MSP:AVG <int1></pre> |
| Parameters | <p><i>int1</i> is an integer. The minimum value for <i>int1</i> is 0, the maximum value is 100. The default value for <i>int1</i> is 0.</p> |
| Description | <p>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: Note To read out the measurement result values of the averaging process, the :FETCh:GPRS:RFTX:BLOCKdata:MSpectrum command may be used.</p> |
| Query | <p>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.</p> |
| Example | <pre>:MEAS:CDMA:CONT:MSpectrum:AVG? 25</pre> <p>The array returned in this example is: "-61.0,-61.0,-61.1, ...0.0,0.0,0.0,...-61.0".</p> |

:MEASure:CDMA[:CONTInuous]:RFRX:FER:IMMediate

| | |
|--------------------|--|
| Syntax | <pre>:MEASure:CDMA[:CONTInuous]:RFRX:FER:IMMediate</pre> |
| Parameters | There are no parameters. |
| Description | Starts a continuous FER measurement. |
| Query | The query form of this command will return a string containing the floating point value with the FER result. |
| Example | <pre>:MEASure:CDMA:RFTX[:CONTInuous]:RFRX:FER:IMMediate?</pre> <p>Starts the measurement and returns the result, e.g. 3.1.</p> |

:MEASure:CDMA[:CONTInuous]:RFRX:RFER:IMMediate

| | |
|-------------------|---|
| Syntax | <pre>:MEASure:CDMA[:CONTInuous]:RFRX:RFER:IMMediate</pre> |
| Parameters | There are no parameters. |

| | |
|--------------------|---|
| Description | Starts a continuous reported FER measurement. |
| Query | The query form of this command will return a floating point value for the result. |
| Example | :MEASure:CDMA:RFTX[:CONTInuous]:RFRX:RFER:IMMediate? |

:MEASure:CDMA[:CONTInuous]:RFRX:RFER:PILA

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFRX:RFER:PILA |
| Parameters | There are no parameters. |
| Description | Starts a continuous reported FER measurement. |
| Query | The query form of this command will return a floating point value. |
| Example | :MEASure:CDMA:RFTX[:CONTInuous]:RFRX:RFER:PILA? Returns a measurement result, e.g. -7.1. |

:MEASure:CDMA[:CONTInuous]:RFRX:RFER:PILB

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA[:CONTInuous]:RFRX:RFER:PILB |
| Parameters | There are no parameters. |
| Description | Starts a continuous reported FER measurement. |
| Query | The query form of this command will return a floating point value. |
| Example | :MEASure:CDMA:RFTX[:CONTInuous]:RFRX:RFER:PILB? |

:MEASure:CDMA:ARRay:RFTX:GRoup

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:GRoup <Int1> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of a group of RF TX tests as many times as specified with Int1. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :CONF:CDMA:MEAS:GRO:RFTX WQU,POW :MEASure:CDMA:ARR:RFTX:GRoup? 10 In this example, the group of measurements is defined by waveform quality and RF power results. Ten measurements are performed and the results returned in a string. |

:MEASure:CDMA:ARRay:RFTX:FREQuency:ERRor

| | |
|-------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:FREQuency:ERRor <Int1> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0. |

| | |
|--------------------|---|
| Description | Performs the frequency error measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:CDMA:ARRay:RFTX:FREQ:ERR? 5 Starts five measurements, the results of which are returned. Example: 47.1, 91.5, 13.6, -4.8, 15.3 |

:MEASure:CDMA:ARRay:RFTX:Power

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA:ARRay:RFTX:Power <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's RF power as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:CDMA:ARR:RFTX:POWER? 10 |

:MEASure:CDMA:ARRay:RFTX:MAXimum

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:MAXimum <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's maximum RF power as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the measurement results. |
| Example | :MEASure:CDMA:ARR:RFTX:MAXimum? 10 |

:MEASure:CDMA:ARRay:RFTX:MINimum

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:MINimum <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's minimum RF power as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARR:RFTX:MINimum? 10 |

:MEASure:CDMA:ARRay:RFTX:STANdby

| | |
|---------------|---|
| Syntax | :MEASure:CDMA:ARRay:RFTX:STANdby <Int1> |
|---------------|---|

| | |
|--------------------|--|
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the standby RF power as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the result (in dBm). |
| Example | :MEASure:CDMA:ARR:RFTX:STANdby? 10 |

:MEASure:CDMA:ARRay:RFTX:ACCess

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA:ARRay:RFTX:ACCess <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the access channel RF power as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results (in dBm). |
| Example | :MEASure:CDMA:ARR:RFTX:ACCess? 1 |

:MEASure:CDMA:ARRay:RFTX:CODEdomain:IPOWer

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA:ARRay:RFTX:CODEdomain:IPOWer <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 1. The default value is 0. |
| Description | Performs a measurement of the code domain I channel power as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results, in dB. |
| Example | :MEASure:CDMA:ARR:RFTX:CODE:IPOW? 1 |

:MEASure:CDMA:ARRay:RFTX:CODEdomain:INOIse

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:CODEdomain:INOIse <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 1. The default value is 0. |
| Description | Performs a measurement of the code domain I channel noise as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results (in dB). |
| Example | :MEASure:CDMA:ARR:RFTX:CODE:INOIse? 1 |

:MEASure:CDMA:ARRay:RFTX:CODE:QPOWer

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:CODE:QPOWer <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 1. The default value is 0. |
| Description | Performs a measurement of the code domain Q channel power. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results (in dB). |
| Example | :MEASure:CDMA:ARR:RFTX:CODE:QPOWer? 1 |

:MEASure:CDMA:ARRay:RFTX:CODE:QNOIse

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:CODE:QNOIse <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 1. The default value is 0. |
| Description | Performs a measurement of the code domain Q channel noise as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results (in dB). |
| Example | :MEASure:CDMA:ARR:RFTX:CODE:QNOIse? 1 |

:MEASure:CDMA:ARRay:RFTX:GATed:[IMMediate]

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA:ARRay:RFTX:GATed:[IMMediate] <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a measurement of the gated power/time template measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and returns the pass/fail results. |
| Example | :MEASure:CDMA:ARR:RFTX:GATed:IMM? 12 |

:MEASure:CDMA:ARRay:RFTX:GATed:AVERAge

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:GATed:AVERAge <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a measurement of the gated power/time template measurement as many times as specified, and gets the average values. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |

| | |
|----------------|--|
| Example | :MEASure:CDMA:ARR:RFTX:GATed:AVERage? 10 |
|----------------|--|

:MEASure:CDMA:ARRay:RFTX:GATed:DROOp

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA:ARRay:RFTX:GATed:DROOp <Int1> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a measurement of the gated power/time template measurement as many times as specified, and get the minimum values. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARR:RFTX:GATed:DROOp? 20 |

:MEASure:CDMA:ARRay:RFTX:GATed:PEAK

| | |
|--------------------|---|
| Syntax | :MEASure:CDMA:ARRay:RFTX:GATed:PEAK <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a measurement of the gated power/time template measurement as many times as specified, and get the maximum values. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARR:RFTX:GATed:PEAK? 30 |

:MEASure:CDMA:ARRay:RFTX:OPEN

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:OPEN <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a measurement of the open loop power a specified number of times. |
| Query | The query form of this command will return a predefined expression: PASS FAIL NONE for each of the measurements. |
| Example | :MEASure:CDMA:ARR:RFTX:OPEN:IMM? 10 |

:MEASure:CDMA:ARRay:RFTX:OPEN:INITial

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:OPEN:INITial <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs an open loop measurement and gets the initial power values, as many times as specified. |

| | |
|----------------|--|
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARR:RFTX:OPEN:INIT? 10 |

:MEASure:CDMA:ARRay:RFTX:CLOSEd

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:CLOSEd <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a measurement of the closed loop power a specified number of times. |
| Query | The query form of this command will return a predefined expression: PASS FAIL NONE for each of the measurements. |
| Example | :MEASure:CDMA:ARR:RFTX:CLOS:IMM? 100 |

:MEASure:CDMA:ARRay:RFTX:CLOSEd:INITial

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:CLOSEd:INITial <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a closed loop measurement and gets the initial power values as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARR:RFTX:CLOS:INIT? 10 |

:MEASure:CDMA:ARRay:RFTX:CLOSEd:MAXimum

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:CLOSEd:MAXimum <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a closed loop measurement and gets the maximum power values as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARR:RFTX:CLOS:MAXimum? 10 |

:MEASure:CDMA:ARRay:RFTX:CLOSEd:MINimum

| | |
|-------------------|---|
| Syntax | :MEASure:CDMA:ARRay:RFTX:CLOSEd:MINimum <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |

| | |
|--------------------|--|
| Description | Performs a closed loop measurement and gets the minimum power values as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARR:RFTX:CLOS:MINimum? 10 |

:MEASure:CDMA:ARRay:RFTX:WQQuality

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:WQQuality <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a waveform quality measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARR:RFTX:WQU? 10 |

:MEAS:CDMA:ARRay:RFTX:CDPHase?

| | |
|--------------------|---|
| Syntax | :MEAS:CDMA:ARRay:RFTX:CDPHase? <Int1>, <PredefExp> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 1. The default value is 0. PredefExp is an optional parameter and one of the predefined expressions ALL RPICh RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. The default is ALL. The channels are: RPICh = R_PICH (W032) = pilot channel RDCCh = R_DCCH (W816) = dedicated control channel RSCH2x2 = R_SCH2_x2 (W68) = supplementary channel 2 RSCH2x4 = R_SCH2_x4 (W24) = supplementary channel 2 RFCH = R_FCH (W416) = fundamental channel RCCCh = R_CCCH (W28) = common control channel RSCH1x4 = R_SCH1_x4 (W24) = supplementary channel 1 RSCH1x8 = R_SCH1_x8 (W12) = supplementary channel 1 |
| Description | The command performs the code domain phase measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results (in radians). |
| Example | :MEASure:CDMA:ARR:RFTX:CDPH? 1, ALL |

:MEAS:CDMA:ARRay:RFTX:CDPower?

| | |
|---------------|--|
| Syntax | :MEAS:CDMA:ARRay:RFTX:CDPower? <Int1>, <PredefExp> |
|---------------|--|

| | |
|--------------------|---|
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 1. The default value is 0. PredefExp is an optional parameter and is one of the predefined expressions ALL RPICh RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. The default is ALL. |
| Description | The command performs the code domain power measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results (in dB). |
| Example | :MEASure:CDMA:RFTX:CDPower? 1,ALL |

:MEAS:CDMA:ARRAY:RFTX:CDTime?

| | |
|--------------------|---|
| Syntax | :MEAS:CDMA:ARRAY:RFTX:CDTime? <Int1>, <PredefExp> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 1. The default value is 0. PredefExp is an optional parameter and is one of the predefined expressions ALL RPICh RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. The default is ALL. |
| Description | The command performs the code domain time measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results (in microseconds). |
| Example | :MEASure:CDMA:RFTX:CDTime? 1,ALL |

:MEASure:CDMA:ARRAY:RFTX:TIME:OFFSet

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRAY:RFTX:TIME:OFFSet <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a time offset measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARRAY:RFTX:TIME:OFFSet? 5 |

:MEASure:CDMA:ARRAY:RFTX:VERRor:RMS

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRAY:RFTX:VERRor:RMS <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs an RMS vector error measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |

Example :MEASure:CDMA:ARRay:RFTX:VERRor:RMS? 19

:MEASure:CDMA:ARRay:RFTX:VERRor:PEAK

Syntax :MEASure:CDMA:ARRay:RFTX:VERRor:PEAK <Int1>
Parameters Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0.
Description Performs a peak vector error measurement as many times as specified.
Query The query form of this command will perform the measurement the specified number of times, and return the results.
Example :MEASure:CDMA:ARRay:RFTX:VERRor:PEAK? 10

:MEASure:CDMA:ARRay:RFTX:VERRor:OOFFset

Syntax :MEASure:CDMA:ARRay:RFTX:VERRor:OOFFset <Int1>
Parameters Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0.
Description Performs an origin offset vector error measurement as many times as specified.
Query The query form of this command will perform the measurement the specified number of times, and return the results.
Example :MEASure:CDMA:ARRay:RFTX:VERRor:OOFFset? 10

:MEASure:CDMA:ARRay:RFTX:VERRor:IQIMbalance

Syntax :MEASure:CDMA:ARRay:RFTX:VERRor:IQIMbalance <Int1>
Parameters Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0.
Description Performs an IQ imbalance vector error measurement as many times as specified.
Query The query form of this command will perform the measurement the specified number of times, and return the results.
Example :MEASure:CDMA:ARRay:RFTX:VERRor:IQIMbalance? 10

:MEASure:CDMA:ARRay:RFTX:VERRor:MAGNitude:RMS

Syntax :MEASure:CDMA:ARRay:RFTX:VERRor:MAGNitude:RMS <Int1>
Parameters Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0.
Description Performs an RMS magnitude vector error measurement as many times as specified.
Query The query form of this command will perform the measurement the specified number of times, and return the results.

Example :MEASure:CDMA:ARRay:RFTX:VERRor:MAGN:RMS? 10

:MEASure:CDMA:ARRay:RFTX:VERRor:MAGNitude:PEAK

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:VERRor:MAGNitude:PEAK <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a peak magnitude vector error measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARRay:RFTX:VERRor:MAGN:PEAK? 10 |

:MEASure:CDMA:ARRay:RFTX:VERRor:PHASe:RMS

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:VERRor:PHASe:RMS <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs an RMS phase vector error measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARRay:RFTX:VERRor:PHase:RMS? 10 |

:MEASure:CDMA:ARRay:RFTX:VERRor:PHASe:PEAK

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFTX:VERRor:PHASe:PEAK <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a peak phase vector error measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:ARRay:RFTX:VERRor:PHase:PEAK? 10 |

:MEAS:CDMA:ARR:RFSP:ACPM:MODulation

| | |
|--------------------|--|
| Syntax | :MEAS[:CDMA]: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). |

| | |
|----------------|---|
| Query | The query form of this command will perform the measurement the specified number of times (<code>int1</code> 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 $(25 * int1)$ floating point real numbers. The physical dimension is dBc. The single measurement results are separated by commas. |
| Example | <code>:MEAS:CDMA:ARR:RFSpectrum:ACPM:MODulation 5</code> <code>:FETCh:CDMA:RFSP:ACPM:MOD</code> The string returned would contain 125 floating point real numbers, representing the 25 measurement result values of five measurement runs. |

:MEASure:CDMA:ARRay:RFRX:FER:IMMediate

| | |
|--------------------|--|
| Syntax | <code>:MEASure:CDMA:ARRay:RFRX:FER:IMMediate <Int1></code> |
| Parameters | <code>Int1</code> is an integer. The minimum value for <code>Int1</code> is 0, the maximum value is 100. The default value is 0. |
| Description | Performs an FER measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | <code>:MEASure:CDMA:RFTX:ARR:RFRX:FER:IMMediate? 100</code> |

:MEASure:CDMA:ARRay:RFRX:RFER:IMMediate

| | |
|--------------------|--|
| Syntax | <code>:MEASure:CDMA:ARRay:RFRX:RFER:IMMediate <Int1></code> |
| Parameters | <code>Int1</code> is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a reported FER measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | <code>:MEASure:CDMA:RFTX:ARR:RFRX:RFER:IMMediate? 10</code> |

:MEASure:CDMA:ARRay:RFRX:RFER:PILA

| | |
|--------------------|--|
| Syntax | <code>:MEASure:CDMA:ARRay:RFRX:RFER:PILA <Int1></code> |
| Parameters | <code>Int1</code> is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a reported FER measurement and sector 1 pilot as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | <code>:MEASure:CDMA:RFTX:ARR:RFRX:RFER:PILA? 10</code> |

:MEASure:CDMA:ARRay:RFRX:RFER:PILB

| | |
|--------------------|--|
| Syntax | :MEASure:CDMA:ARRay:RFRX:RFER:PILB <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs a reported FER measurement and sector 2 pilot as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times, and return the results. |
| Example | :MEASure:CDMA:RFTX:ARR:RFRX:RFER:PILB? 30 |

:MEASure:AMPS[:CONTinuous]:RFTX:GROup

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS[:CONTinuous]:RFTX:GROup |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the RF TX tests. |
| Query | The query form of this command will start the sequence of RF TX measurements. |
| Example | :CONF:AMPS:MEAS:GRO:RFTX SATD, POW :MEASure:AMPS:RFTX:GROup? In this example, the group of measurements is defined by SAT Deviation and RF power results. |

:MEASure:AMPS[:CONTinuous]:RFTX:POWer

| | |
|--------------------|--|
| Syntax | :MEASure:AMPS[:CONTinuous]:RFTX:POWer |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the mobile's RF power. |
| Query | The query form of this command will return the float value for power. |
| Example | :MEASure:AMPS:RFTX:POWer? Starts an RF power measurement and returns the result (in dBm), e.g. -45.3. |

:MEASure:AMPS[:CONTinuous]:RFTX:DEV:SAT

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS[:CONTinuous]:RFTX:DEV:SAT |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Starts a continuous measurement of the mobile's SAT deviation. |
| Query | The query form of this command will return the float value for deviation. |
| Example | :MEASure:AMPS:RFTX:DEV:SAT? |

:MEASure:AMPS[:CONTInuous]:RFTX:DEV:ST

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS[:CONTInuous]:RFTX:DEV:ST |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the mobile's ST deviation. |
| Query | The query form of this command will return the float value for deviation. |
| Example | :MEASure:AMPS:RFTX:DEV:ST? |

:MEASure:AMPS[:CONTInuous]:RFTX:DEV:AUD

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS[:CONTInuous]:RFTX:DEV:AUD |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Starts a continuous measurement of the mobile's Audio deviation. |
| Query | The query form of this command will return the float value for deviation. |
| Example | :MEASure:AMPS:RFTX:DEV:AUD? |

:MEASure:AMPS[:CONTInuous]:RFTX:FREQuency:ERRor:RF

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS[:CONTInuous]:RFTX:FREQuency:ERRor:RF |
| Parameters | There are no parameters. |
| Description | Starts a continous measurement of the frequency error. |
| Query | The query form of this command will start the sequence of frequency error measurements. |
| Example | :MEASure:AMPS:RFTX:FREQ:ERR:RF? Starts a frequency error measurement and returns the result. |

:MEASure:AMPS[:CONTInuous]:RFTX:FREQ:ERR:SAT

| | |
|--------------------|--|
| Syntax | :MEASure:AMPS[:CONTInuous]:RFTX:FREQ:ERR:SAT |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Starts a continuous measurement of the frequency error. |
| Query | The query form of this command will start the sequence of frequency error measurements. |
| Example | :MEASure:AMPS:RFTX:FREQ:ERR:SAT? Starts a frequency error measurement and returns the result. |

:MEASure:AMPS[:CONTinuous]:RFTX:FREQ:ERR:ST

| | |
|--------------------|--|
| Syntax | :MEASure:AMPS[:CONTinuous]:RFTX:FREQ:ERR:ST |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Starts a continuous measurement of the frequency error. |
| Query | The query form of this command will start the sequence of frequency error measurements. |
| Example | :MEASure:AMPS:RFTX:FREQ:ERR:ST? Starts a frequency error measurement and returns the result.. |

:MEASure:AMPS[:CONTinuous]:RFRX:GRoup

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS[:CONTinuous]:RFRX:GRoup |
| Parameters | There are no parameters |
| Description | Starts a continuous measurement of the RFRX measurement group. |
| Query | The query form of this command will start the sequence of measurements. |
| Example | :CONF:AMPS:MEAS:GRO:RFRX SIN,ACL :MEASure:AMPS:RFTX:GRoup? In this example, the group of measurements is defined by Sinad and AC Level results. |

:MEASure:AMPS[:CONTinuous]:RFRX:SINad

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS[:CONTinuous]:RFRX:SINad |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the Sinad measurement. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:AMPS:RFTX:SINad? |

:MEASure:AMPS[:CONTinuous]:RFRX:ACLevel

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS[:CONTinuous]:RFRX:ACLevel |
| Parameters | There are no parameters. |
| Description | Starts a continuous measurement of the AC Level measurement. |
| Query | The query form of this command will start the sequence of measurements. |
| Example | :MEASure:AMPS:RFTX:ACLevel? |

:MEASure:AMPS:ARRay:RFTX:GROup

| | |
|--------------------|--|
| Syntax | :MEASure:AMPS:ARRay:RFTX:GROup <Int1> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of a group of RF TX tests as many times as specified with Int1. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :CONF:AMPS:MEAS:GRO:RFTX WQU,POW :MEASure:AMPS:ARR:RFTX:GROup? 10 In this example, the group of measurements is defined by waveform quality and RF power results. Ten measurements are performed and the results returned in a string. |

:MEASure:AMPS:ARRay:RFTX:Power

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS:ARRay:RFTX:Power <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's RF power as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:AMPS:ARR:RFTX:POWer? 10 |

:MEASure:AMPS:ARRay:RFTX:DEV:SAT

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS:ARRay:RFTX:DEV:SAT <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's SAT deviation as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:AMPS:ARR:RFTX:DEV:SAT? 10 |

:MEASure:AMPS:ARRay:RFTX:DEV:ST

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS:ARRay:RFTX:DEV:ST <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's ST deviation as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |

Example :MEASure:AMPS:ARR:RFTX:DEV:ST? 10

:MEASure:AMPS:ARRay:RFTX:DEV:AUD

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS:ARRay:RFTX:DEV:AUD <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's Audio deviation as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:AMPS:ARR:RFTX:DEV:AUD? 10 |

:MEASure:AMPS:ARRay:RFTX:FREQuency:ERRor:RF

| | |
|--------------------|--|
| Syntax | :MEASure:AMPS:ARRay:RFTX:FREQuency:ERRor:RF <Int1> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the frequency error measurement as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:AMPS:ARRay:RFTX:FREQ:ERR:RF? 5 Starts five measurements, the results of which are returned. Example: 47.1, 91.5, 13.6, -4.8, 15.3 |

:MEASure:AMPS:ARRay:RFTX:FREQ:ERR:SAT

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS:ARRay:RFTX:FREQ:ERR:SAT <Int1> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's SAT frequency error as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:AMPS:ARR:RFTX:FREQ:ERR:SAT? 10 |

:MEASure:AMPS:ARRay:RFTX:FREQ:ERR:ST

| | |
|-------------------|--|
| Syntax | :MEASure:AMPS:ARRay:RFTX:FREQ:ERR:ST <Int1> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0. |

| | |
|--------------------|---|
| Description | Performs the measurement of the mobile's ST frequency error as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:AMPS:ARR:RFTX:FREQ:ERR:ST? 10 |

:MEASure:AMPS:ARRay:RFRX:GRoup

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS:ARRay:RFRX:GRoup <Int1> |
| Parameters | Int1 is an integer. The minimum value for Int1 is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of a group of RF TX tests as many times as specified with Int1. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :CONF:AMPS:MEAS:GRO:RFRX SATD, POW :MEASure:AMPS:ARR:RFTX:GRoup? 10 In this example, the group of measurements is defined by SAT deviation and RF power results. Ten measurements are performed and the results returned in a string. |

:MEASure:AMPS:ARRay:RFRX:SINad

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS:ARRay:RFRX:SINad <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's sinad as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:AMPS:ARR:RFTX:SINad? 10 |

:MEASure:AMPS:ARRay:RFRX:ACLevel

| | |
|--------------------|---|
| Syntax | :MEASure:AMPS:ARRay:RFRX:ACLevel <Int1> |
| Parameters | Int1 is an integer. The minimum value is 0, the maximum value is 100. The default value is 0. |
| Description | Performs the measurement of the mobile's AC level as many times as specified. |
| Query | The query form of this command will perform the measurement the specified number of times and return the results. |
| Example | :MEASure:AMPS:ARR:RFTX:ACLevel? 10 |

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 | <p>:MEAS:GSM:RFTX:PRMS :MEAS:GSM:RFRX:RBER:ALL :FETCh:LAST?</p> <p>In this example, the FETCh:LAST command will return the three residual bit error ratios (three floating point real numbers) as this is the measurement started last.</p> |

: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 | <p>:MEAS:BLOC:AFSP :FETCh:BLOCK:AFSP?</p> <p>The FETCh 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.</p> |

:FETCh:AFANalyser:GROUp?

| | |
|--------------------|---|
| Syntax | :FETCh:AFANalyser:GROUp? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |

| | |
|----------------|--|
| Query | <p>The query form of this command will return the latest measurement result values of the sequence of AF measurements as specified with the <code>:CONF:MEAS:GRO:AFAN</code> command and measured with the <code>:MEAS[ARRay]:AFAN:GRO</code> 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 <code>:CONF:MEAS:GRO:AFAN</code> command. The order of the measurement result values within the result string is as described below for the <code>:FETCh:AFANalyser:ALL</code> command.</p> |
| Example | <pre>:CONF:MEAS:GRO:AFAN SIN,FREQ :MEAS:AFAN:GRO :FETCh:AFAN:GRO?</pre> <p>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.</p> |

:FETCh:AFANalyser:ALL?

| | |
|--------------------|--|
| Syntax | <code>:FETCh:AFANalyser:ALL?</code> |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | <p>The query form of this command will return the latest measurement result values of the <code>:MEAS[:ARRay]:AFAN:ALL</code> measurement in a string. All measurement result values of the AF measurements will be floating point real numbers, separated by commas.</p> <p>The order of the measurement result values within the result string is as follows:</p> <ol style="list-style-type: none"> 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. <p>Note: Any AF test will need the audio option to be installed on your Willtek 4400.</p> |
| Example | <pre>:MEAS:ARR:AFAN:ALL 2 :FETCh:AFAN:ALL?</pre> <p>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".</p> |

: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:DIS 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:DIS :FETC:AFAN:DIS? 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 | <p>This query returns the latest measurement result values of the sequence of power/ current consumption measurements as specified with the <code>:CONF:MEAS:GRO:PSUP</code> command and measured with the <code>:MEAS[ARRAY]:PSUP:GRO</code> 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 <code>:CONF:MEAS:GRO:PSUP</code> command.</p> <p>The order of the measurement result values within the result string is as described below for the <code>:FETCh:PSUP:ALL</code> command.</p> |
| Example | <pre><code>:CONF:MEAS:GRO:PSUP ACUR,APOW :MEAS:PSUP:GRO :FETCh:PSUP:GRO?</code></pre> <p>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".</p> <p>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).</p> |

:FETCh:PSUPply:ALL

| | |
|--------------------|---|
| Syntax | <code>:FETCh:PSUPply:ALL?</code> |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | <p>This query returns the latest measurement result values of the <code>:MEAS[:ARRAY]:PSUP:ALL</code> measurement in a string. All measurement result values of the power and current consumption measurements are floating point real numbers, separated by commas.</p> <p>The order of the measurement result values within the result string is as follows:</p> <ol style="list-style-type: none"> 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. <p>Note: Any current measurement will need the MS Power Supply and Current Measurement options to be installed on your Willtek 4400.</p> |
| Example | <pre><code>:MEAS:PSUP:ALL :FETCh:PSUP:ALL?</code></pre> <p>Returns a string like "893.5,395.4,1256.2".</p> |

:FETCh:PSUPply:APOWer

| | |
|--------------------|---|
| Syntax | <code>:FETCh:PSUPply:APOWer?</code> |
| 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: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 <i>int1</i> times the number of active uplink slots followed by 23 float values per time slot. <i>int1</i> 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 <i>int1</i> 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:TRAN-sient. |
| Example | :CONF:GPRS:MSTA:MSL 2 :FETCh:GPRS:MULT:RFSP:ACPM:TRAN? |

:FETCh:CDMA:LAST

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:LAST |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | Returns exactly the same values returned by the last FETCh or MEASure query. |
| Example | :FETCh:CDMA:LAST? |

:FETCh:CDMA:RFTX:GROup

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:GROup |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | Returns the same parameters as for the :MEASure:ARRAY:RFTX:GROup measurement and as defined by :CONFigure:MEASure:GROup:RFTX. |
| Example | :FETCh:CDMA:RFTX:GRO? |

:FETCh:CDMA:RFTX:FREQ:ERRor

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:FREQ:ERRor? |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:CDMA:CONT ARRay:RFTX:FREQ:ERR measurement command. |
| Example | :FETCh:CDMA:RFTX:FREQ? |

:FETCh:CDMA:RFTX:POWer

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:POWer? |
| Parameters | There are no parameters. |
| Description | The query form of this command is supported only. |
| Query | The command returns the results from the :MEASure:CDMA:CONT ARRay:RFTX:POWer measurement command. |
| Example | :FETCh:CDMA:RFTX:POW? |

:FETCh:CDMA:RFTX:MAXimum

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:MAXimum? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the results from the :MEASure:CDMA:CONT ARRay:RFTX:MAXimum measurement command. |
| Example | :FETCh:CDMA:RFTX:MAXimum? |

:FETCh:CDMA:RFTX:MINimum

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:MINimum? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the results from the :MEASure:CDMA:CONT ARRay:RFTX:MINimum measurement command. |
| Example | :FETCh:CDMA:RFTX:MINimum? |

:FETCh:CDMA:RFTX:STANdby

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:STANdby? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the results from the :MEASure:CDMA:CONT ARRay:RFTX:STANdby measurement command. |
| Example | :FETCh:CDMA:RFTX:STANdby? |

:FETCh:CDMA:RFTX:ACCess

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:ACCess? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The commands return the results from the :MEA-Sure:CDMA:CONT:RFTX:ACCess measurement command. |
| Example | :FETCh:CDMA:RFTX:ACCess? |

:FETCh:CDMA:RFTX:CODEdomain:IMMediate

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:CODEdomain:IMMediate? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns one of the predefined expressions PASS FAIL NONE as a result from a previous code domain measurement command. |
| Example | :FETCh:CDMA:RFTX:CODEdomain? |

:FETCh:CDMA:RFTX:CODEdomain:IPOWER?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:CODEdomain:IPOWER? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the 16 results from a previous :MEASure:CDMA:CONT:RFTX:CODE:IPOW measurement command. |
| Example | :FETCh:CDMA:RFTX:CODE:IPOWER? |

:FETCh:CDMA:RFTX:CODEdomain:INOIse

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:CODEdomain:INOIse? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the 16 results from a previous :MEASure:CDMA:CONT:RFTX:CODE:INOIse measurement command. |
| Example | :FETCh:CDMA:RFTX:CODE:INOIse? |

:FETCh:CDMA:RFTX:CODEdomain:QPOWER?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:CODEdomain:QPOWER? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The commands return the 16 results from a previous :MEASure:CDMA:CONT:RFTX:CODE:QPOWER measurement command. |
| Example | :FETCh:CDMA:RFTX:CODE:QPOWER? |

:FETCh:CDMA:RFTX:CODEdomain:QNOIse?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:CODEdomain:QNOIse? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the 16 results from a previous :MEASure:CDMA:CONT:RFTX:CODE:QNOIse measurement command. |
| Example | :FETCh:CDMA:RFTX:CODE:QNOIse? |

:FETCh:CDMA:RFTX:GATed:IMMediate?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:GATed:IMMediate? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns one of the predefined expressions PASS FAIL NONE as result from a gated power measurement command. |
| Example | :FETCh:CDMA:RFTX:GATed:IMMediate? |

:FEtCh:CDMA:RfTX:GATed:AVERAge?

| | |
|--------------------|---|
| Syntax | :FEtCh:CDMA:RfTX:GATed:AVERAge? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RfTX:GATed:AVERAge measurement command. |
| Example | :FEtCh:CDMA:RfTX:GATed:AVERAge? |

:FEtCh:CDMA:RfTX:GATed:MINimum

| | |
|--------------------|---|
| Syntax | :FEtCh:CDMA:RfTX:GATed:MINimum? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RfTX:GATed:MINimum measurement command. |
| Example | :FEtCh:CDMA:RfTX:GATed:MINimum? |

:FEtCh:CDMA:RfTX:GATed:MAXimum?

| | |
|--------------------|---|
| Syntax | :FEtCh:CDMA:RfTX:GATed:MAXimum? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The commands return result from a previous :MEASure:CDMA:CONT:RfTX:GATed:MAXimum measurement command. |
| Example | :FEtCh:CDMA:RfTX:GATed:MAXimum? |

:FEtCh:CDMA:RfTX:OPEN:IMMediate?

| | |
|--------------------|---|
| Syntax | :FEtCh:CDMA:RfTX:OPEN:IMMediate? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns one of the predefined expressions PASS FAIL NONE as result from an open loop power measurement command. |
| Example | :FEtCh:CDMA:RfTX:OPEN:IMMediate? |

:FETCh:CDMA:RFTX:OPEN:INITial

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:OPEN:INITial? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:OPEN:INITial measurement command. |
| Example | :FETCh:CDMA:RFTX:OPEN:INITial? |

:FETCh:CDMA:RFTX:CLOSeD:IMMediate?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:CLOSeD:IMMediate? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns one of the predefined expressions PASS FAIL NONE as a result from a closed loop power measurement command. |
| Example | :FETCh:CDMA:RFTX:CLOSeD:IMMediate? |

:FETCh:CDMA:RFTX:CLOSeD:INITial?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:CLOSeD:INITial? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:CLOSeD:INITial measurement command. |
| Example | :FETCh:CDMA:RFTX:CLOSeD:INITial? |

:FETCh:CDMA:RFTX:CLOSeD:MAXimum?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:CLOSeD:MAXimum? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:CLOSeD:MAXimum measurement command. |
| Example | :FETCh:CDMA:RFTX:CLOSeD:MAXimum? |

:FETCh:CDMA:RFTX:CLOSeD:MINimum?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:CLOSeD:MINimum? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:CLOSeD:MINimum measurement command. |
| Example | :FETCh:CDMA:RFTX:CLOSeD:MINimum? |

:FETCh:CDMA:RFTX:WQUality?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:WQUality? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:WQUality measurement command. |
| Example | :FETCh:CDMA:RFTX:WQUality? |

:FETCh:CDMA:RFTX:CDPHase?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:CDPHase? <PredefExp> |
| Parameters | PredefExp is one of the predefined expressions ALL RPICh RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. The default is ALL. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:CDPHase measurement command. |
| Example | :FETCh:CDMA:RFTX:CDPHase? |

:FETCh:CDMA:RFTX:CDPower?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:CDPower? <PredefExp> |
| Parameters | PredefExp is one of the predefined expressions ALL RPICh RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. The default is ALL. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:CDPower measurement command. |
| Example | :FETCh:CDMA:RFTX:CDPower? |

:FETCh:CDMA:RFTX:CDTIme?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:CDTIme? <PredefExp> |
| Parameters | PredefExp is one of the predefined expressions ALL RPICh RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. The default is ALL. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:CDTIme measurement command. |
| Example | :FETCh:CDMA:RFTX:CDTIme? |

:FETCh:CDMA:RFTX:TIme

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:TIme? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:TIme measurement command. |
| Example | :FETCh:CDMA:RFTX:TIme? |

:FETCh:CDMA:RFTX:VERRor:RMS?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:VERRor:RMS? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:VERR:RMS measurement command. |
| Example | :FETCh:CDMA:RFTX:VERRor:RMS? |

:FETCh:CDMA:RFTX:VERRor:PEAK?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:VERRor:PEAK? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:VERR:PEAK measurement command. |
| Example | :FETCh:CDMA:RFTX:VERRor:PEAK? |

:FETCh:CDMA:RFTX:VERRor:OOffset

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:VERRor:OOffset? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:VERR:OOffset measurement command. |
| Example | :FETCh:CDMA:RFTX:VERRor:OOffset? |

:FETCh:CDMA:RFTX:VERRor:IQIMbalance?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:VERRor:IQIMbalance? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:VERR:IQIMbalance measurement command. |
| Example | :FETCh:CDMA:RFTX:VERRor:IQIMbalance? |

:FETCh:CDMA:RFTX:VERRor:MAGNitude:RMS

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:VERRor:MAGNitude:RMS? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:VERR:MAGNitude:RMS measurement command. |
| Example | :FETCh:CDMA:RFTX:VERRor:MAGNitude:RMS? |

:FETCh:CDMA:RFTX:VERRor:MAGNitude:PEAK?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:VERRor:MAGNitude:PEAK? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:VERR:MAGNitude:PEAK measurement command. |
| Example | :FETCh:CDMA:RFTX:VERRor:MAGNitude:PEAK? |

:FETCh:CDMA:RFTX:VERRor:PHASe:RMS

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFTX:VERRor:PHASe:RMS? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:VERR:PHASE:RMS measurement command. |
| Example | :FETCh:CDMA:RFTX:VERRor:PHASE:RMS? |

:FETCh:CDMA:RFTX:VERRor:PHASe:PEAK

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFTX:VERRor:PHASe:PEAK? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFTX:VERR:PHASE:PEAK measurement command. |
| Example | :FETCh:CDMA:RFTX:VERRor:PHASE:PEAK? |

:FETCh:CDMA:RFRX:FER:IMMediate?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFRX:FER:IMMediate? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFRX:FER:IMMediate measurement command. |
| Example | :FETCh:CDMA:RFRX:FER:IMMediate? |

:FETCh:CDMA:RFRX:FER:FRAME:ERRor

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFRX:FER:FRAME:ERRor? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFRX:FER:FRAME:ERRor measurement command. |
| Example | :FETCh:CDMA:RFRX:FER:FRAME:ERR? |

:FETCh:CDMA:RFRX:FER:FRAMe:TRANsmitted?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFRX:FER:FRAMe:TRANsmitted? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFRX:FER:FRAMe:TRANsmitted measurement command. |
| Example | :FETCh:CDMA:RFRX:FER:FRAMe:TRANsmitted? |

:FETCh:CDMA:RFRX:FER:STATus:IMMediate

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFRX:FER:STATus:IMMediate? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns one of the predefined expressions PASS FAIL CONPASS CONFALL NONE as result from a FER measurement command. |
| Example | :FETCh:CDMA:RFRX:FER:STAT:IMMediate? |

:FETCh:CDMA:RFRX:RFER:IMMediate

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFRX:RFER:IMMediate? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFRX:RFER:IMMediate measurement command. |
| Example | :FETCh:CDMA:RFRX:RFER:IMMediate? |

:FETCh:CDMA:RFRX:RFER:PILA:LEVel?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFRX:RFER:PILA:LEVel? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFRX:RFER:PILA:LEVel measurement command. |
| Example | :FETCh:CDMA:RFRX:RFER:PILA:LEVel? |

:FETCh:CDMA:RFRX:RFER:PILB:LEVel

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA:RFRX:RFER:PILB:LEVel? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFRX:RFER:PILB:LEVel measurement command. |
| Example | :FETCh:CDMA:RFRX:RFER:PILB:LEVel? |

:FETCh:CDMA:RFRX:RFER:FRAMe:ERRor?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFRX:RFER:FRAMe:ERRor? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFRX:RFER:FRAMe:ERRor measurement command. |
| Example | :FETCh:CDMA:RFRX:RFER:FRAMe:ERR? |

:FETCh:CDMA:RFRX:RFER:FRAMe:TRANSmitted?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA:RFRX:RFER:FRAMe:TRANSmitted? |
| Parameters | There are no parameters. |
| Description | The query form is supported only. |
| Query | The command returns the result from a previous :MEASure:CDMA:CONT:RFRX:RFER:FRAMe:TRANSmitted measurement command. |
| Example | :FETCh:CDMA:RFRX:RFER:FRAMe:TRANSmitted? |

:FETCh:CDMA:RFSP:ACPM:MODulation?

| | |
|--------------------|---|
| Syntax | :FETCh:CDMA: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. |
| Example | :MEAS:CDMA:ARRay:RFSPectrum:ACPM:MOD 5 :FETC:CDMA:RFSP:ACPM:MOD? The Fetch command will return a data array, containing 5 times 25 floating point real numbers. |

:FETCh:CDMA:MSpectrum?

| | |
|--------------------|--|
| Syntax | :FETCh:CDMA: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:CDMA: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:CDMA:MSP :FETCh:CDMA:MSp? The FETCh 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:AMPS:LAST

| | |
|--------------------|--|
| Syntax | :FETCh:AMPS:LAST |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | Returns exactly the same values returned by the last FETCh: or MEASure: query. |
| Example | :FETCh:AMPS:LAST? |

:FETCh:AMPS:RFTX:GRoup

| | |
|--------------------|---|
| Syntax | :FETCh:AMPS:RFTX:GRoup |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | Returns the same parameters as for the :MEASure:ARRAY:RFTX:GRoup measurement and as defined by :CONFIgure:MEASure:GRoup:RFTX. |
| Example | :FETCh:AMPS:RFTX:GRo? |

:FETCh:AMPS:RFTX:POWer

| | |
|--------------------|---|
| Syntax | :FETCh:AMPS:RFTX:POWer? |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:AMPS:CONT ARRAy:RFTX:POWer measurement command. |

| | |
|----------------|-----------------------|
| Example | :FETCh:AMPS:RFTX:POW? |
|----------------|-----------------------|

:FETCh:AMPS:RFTX:DEV:SAT

| | |
|--------------------|--|
| Syntax | :FETCh:AMPS:RFTX:DEV:SAT <Int1> |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:AMPS:CONT ARRay:RFTX:DEV:SAT measurement command. |
| Example | :FETCh:AMPS:RFTX:DEV:SAT? |

:FETCh:AMPS:RFTX:DEV:ST

| | |
|--------------------|---|
| Syntax | :FETCh:AMPS:RFTX:DEV:ST <Int1> |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:AMPS:CONT ARRay:RFTX:DEV:ST measurement command. |
| Example | :FETCh:AMPS:RFTX:DEV:ST? |

:FETCh:AMPS:RFTX:DEV:AUD

| | |
|--------------------|--|
| Syntax | :FETCh:AMPS:RFTX:DEV:AUD <Int1> |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:AMPS:CONT ARRay:RFTX:DEV:AUD measurement command. |
| Example | :FETCh:AMPS:RFTX:DEV:AUD? |

:FETCh:AMPS:RFTX:FREQ:ERRor

| | |
|--------------------|---|
| Syntax | :FETCh:AMPS:RFTX:FREQ:ERRor? |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:AMPS:CONT ARRay:RFTX:FREQ:ERR measurement command. |
| Example | :FETCh:AMPS:RFTX:FREQ? |

:FETCh:AMPS:RFTX:FREQ:ERR:SAT

| | |
|--------------------|--|
| Syntax | :FETCh:AMPS:RFTX:FREQ:ERR:SAT <Int1> |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:AMPS:CONT ARRay:RFTX:FREQ:ERR:SAT measurement command. |
| Example | :FETCh:AMPS:RFTX:FREQ:ERR:SAT? |

:FETCh:AMPS:RFTX:FREQ:ERR:ST

| | |
|--------------------|---|
| Syntax | :FETCh:AMPS:RFTX:FREQ:ERR:ST <Int1> |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:AMPS:CONT ARRay:RFTX:FREQ:ERR:ST measurement command. |
| Example | :FETCh:AMPS:RFTX:FREQ:ERR:ST? |

:FETCh:AMPS:RFRX:GRoup

| | |
|--------------------|---|
| Syntax | :FETCh:AMPS:RFRX:GRoup |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | Returns the same parameters as for the :MEASure:ARRAY:RFRX:GRoup measurement and as defined by :CONFIgure:MEASure:GRoup:RFRX. |
| Example | :FETCh:AMPS:RFRX:GRO? |

:FETCh:AMPS:RFRX:SINad

| | |
|--------------------|--|
| Syntax | :FETCh:AMPS:FRX:SINad <Int1> |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:AMPS:CONT ARRay:RFRX:SINT measurement command. |
| Example | :FETCh:AMPS:RFRX:SINad? |

:FETCh:AMPS:RFRX:ACLevel

| | |
|--------------------|---|
| Syntax | :FETCh:AMPS:RFRX:ACLevel <Int1> |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the :MEASure:AMPS:CONT ARRay:RFRX:ACL measurement command. |
| Example | :FETCh:AMPS:RFTX:ACLevel? |

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 297. |
| Query | There is no query form of this command available. |
| Example | <p>:CALCulate:RESet :MEAS:GSM:ARR:RFTX:LENG 10 :CALC:MAVaverage?</p> <p>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).</p> |

: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 | <p>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.</p> <p>The limits for the single measurements can be set using the appropriate commands of the CALC subsystem (as explained in this section).</p> |

| | |
|----------------|---|
| Example | <pre>:CALCulate:RESet :MEAS:GSM:RFTX:ALL :CALC:LIM:FAIL?</pre> <p>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.</p> |
|----------------|---|

:CALCulate:LIMit:FAIL:CUMulative?

| | |
|--------------------|--|
| Syntax | <code>:CALCulate:LIMit:FAIL:CUMulative?</code> |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | <p>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.</p> <p>The limits for the single measurements can be set using the appropriate commands of the CALC subsystem (as explained in this section).</p> <p>To reset the cumulative check of the measurement results, use the :CALC:LIM:FAIL:CUM:RES command as explained below.</p> |
| Example | <pre>:CALC:LIM:FAIL:CUM:RES :MEAS:GSM:RFTX:TEMP ...(other SCPI commands)...:CALC:LIM:FAIL:CUM?</pre> <p>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.</p> |

:CALCulate:LIMit:FAIL:RESet

| | |
|--------------------|--|
| Syntax | <code>:CALCulate:LIMit:FAIL:RESet</code> |
| 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 | <pre>:CALC:LIM:FAIL:RES :MEAS:GSM:RFTX:TEMP ...(other SCPI commands)...:CALC:LIM:FAIL:CUM?</pre> <p>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.</p> |

: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 | <pre>:CONF:MEAS:GRO:RFTX POW,PRMS :CALC:RES :MEAS:GSM:ARR:RFTX:GROup 20 :CALCulate:MAverage?</pre> <p>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".</p> |

: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 | <pre>:MEAS:AFAN:SIN :CALC:RES ...(other SCPI commands)... :CALCulate:MMIN?</pre> <p>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".</p> |

: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 | <p>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.</p> <p>The position of the boolean number within the string returned indicates the result of the check of following measurements:</p> <ol style="list-style-type: none"> 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. <p>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.</p> |
| Example | <pre>: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.</pre> |

: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 | <pre>:CALC:AFAN:ALL:LIM:STAT ON Switches the limit check of the main audio measurements on.</pre> |

:CALCulate:AFANalyser:ALL:LIMit:UPPer[:DATA]

| | |
|-------------------|--|
| Syntax | :CALCulate:AFANalyser:ALL:LIMit:UPPer[:DATA] <real1>, <real2>, <real3>, <real4>, <real5>, <real6> |
| Parameters | <p>realx are six floating point real numbers.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> |

| | |
|--------------------|--|
| Description | <p>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:</p> <p>real1 represents the upper limit of the peak-to-peak AC voltage measurement; the physical dimension of the number stated is V(pp).</p> <p>real2 represents the upper limit of the RMS-valued AC voltage measurement; the physical dimension of the number stated is V(rms).</p> <p>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).</p> <p>real4 represents the upper limit of the audio frequency measurement; the physical dimension of the number stated is Hertz.</p> <p>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.</p> <p>real6 represents the upper limit of the SINAD signal-to-noise ratio measurement; the physical dimension of the number stated is dB.</p> |
| Query | The query form of this command is not supported. |
| Example | <p>:CALC:AFAN:ALL:LIM:UPP 3,3,1,4000,5,100</p> <p>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.</p> |

:CALCulate:AFANalyser:ALL:LIMit:LOWer[:DATA]

| | |
|--------------------|--|
| Syntax | <pre>:CALCulate:AFANalyser:ALL:LIMit:LOWer[:DATA] <real1>,<real2>,<real3>,<real4>,<real5>,<real6></pre> |
| Parameters | <p>realx are six floating point real numbers.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> |
| Description | <p>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:</p> <p>real1 represents the lower limit of the peak-to-peak AC voltage measurement; the physical dimension of the number stated is V(pp).</p> <p>real2 represents the lower limit of the RMS-valued AC voltage measurement; the physical dimension of the number stated is V(rms).</p> <p>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).</p> <p>real4 represents the lower limit of the audio frequency measurement; the physical dimension of the number stated is Hertz.</p> <p>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.</p> <p>real6 represents the lower limit of the SINAD signal-to-noise ratio measurement; the physical dimension of the number stated is dB.</p> |

| | |
|----------------|---|
| 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:AFANalyser
:ACVoltage:PEAKp:LIMit:LOWer[:DATA]

Syntax **:CALCulate:AFANalyser**
 :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 | <pre>: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.</pre> |
|----------------|--|

:CALCulate:AFANalyser
:DCVoltage:LIMit:STATe

| | |
|--------------------|--|
| Syntax | <pre>:CALCulate:AFANalyser :DCVoltage:LIMit:STATe <PredefExp></pre> |
| 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 | <pre>:CALC:AFAN:DCV:LIM:STAT ON Switches the limit check of the RMS-valued AC voltage ripple measurement of a DC voltage on.</pre> |

:CALCulate:AFANalyser
:DCVoltage:LIMit:UPPer[:DATA]

| | |
|--------------------|---|
| Syntax | <pre>:CALCulate:AFANalyser :DCVoltage:LIMit:UPPer[:DATA] <real1></pre> |
| 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 | <pre>:CALC:AFAN:DCV:LIM:UPP 6 Sets the upper limit of the rms-valued ripple voltage measurement to 6 V.</pre> |

:CALCulate:AFANalyser
:DCVoltage:LIMit:LOWer[:DATA]

| | |
|--------------------|--|
| Syntax | <pre>:CALCulate:AFANalyser :DCVoltage:LIMit:LOWer[:DATA] <real1></pre> |
| 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 | <p>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.</p> <p>The limits can be set using the commands described below.</p> <p>Notes</p> <ul style="list-style-type: none"> - 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 | <pre>:MEAS:ARRay:AFAN:SIN 5 :CALC:AFAN:SIN:LIM?</pre> <p>String returned: "1"</p> <p>This string delivered back indicates that there was a violation of the limits set.</p> |

```
:CALCulate:AFANalyser
:SINad:LIMit:STATe
```

| | |
|--------------------|---|
| Syntax | <pre>:CALCulate:AFANalyser :SINad:LIMit:STATe <PredefExp></pre> |
| 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 | <pre>:CALC:AFAN:SIN:LIM:STAT ON</pre> <p>Switches the limit check of the SINAD measurement on.</p> |

```
:CALCulate:AFANalyser
:SINad:LIMit:UPPer[:DATA]
```

| | |
|--------------------|--|
| Syntax | <pre>:CALCulate:AFANalyser :SINad:LIMit:UPPer[:DATA] <real1></pre> |
| 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 | <pre>:CALC:AFAN:SIN:LIM:UPP 100</pre> <p>Sets the upper limit of the SINAD measurement to 100%nbsp;dB.</p> |

```
:CALCulate:AFANalyser
:SINad:LIMit:LOWer[:DATA]
```

| | |
|---------------|--|
| Syntax | <pre>:CALCulate:AFANalyser :SINad:LIMit:LOWer[:DATA] <real1></pre> |
|---------------|--|

| | |
|--------------------|---|
| Parameters | real1 is a floating point real number. The minimum value is 0.0 , the maximum is 100.0 . The minimum resolution possible is 0.1 , the default value is 0.0 . |
| Description | Sets the lower limit for the SINAD measurement. The physical dimension of the number stated is dB. |
| Query | The query form of this command is not available. |
| Example | :CALC:AFAN:SIN:LIM:LOW 30 Sets the lower limit of the SINAD measurement to 30 dB. |

:CALCulate:AFSPectrum:TEMPlate:LIMit[:FAIL]

| | |
|--------------------|--|
| Syntax | :CALCulate:AFSPectrum:TEMPlate:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers ten boolean numbers, indicating whether the user-definable limits of the audio spectrum measurement were violated by a measurement result value or not. A 0 means that the measurement result values for the respective section of the template were within the limits set, while a 1 indicates that at least one measurement result value did violate at least one of the limits. The limits can be set using the commands described below. |
| Example | :MEAS:ARRay:AFSP 10 :CALC:AFSP:TEMP:LIM:FAIL? String returned: "0,1,0,0,0,0,0,0,0,0" This string delivered back indicates that there was a violation of the limits set for the second area. |

:CALCulate:AFSPectrum:TEMPlate:LIMit:STATe

| | |
|--------------------|--|
| Syntax | :CALCulate:AFSPectrum:TEMPlate:LIMit:STATe <PreDefExp> |
| Parameters | PreDefExp is one of the following predefined expressions: ON OFF . Default is ON . |
| Description | This command switches the check of the measurement result values of the MEAS:AFSP:TEMPlate and MEAS:ARR:AFSP:TEMPlate measurements against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:GSM:RFTX:TEMP:LIM:STAT ON Switches on the limit check of the audio spectrum measurement against the user-defined template. |

:CALCulate:AFSPectrum:TEMPlate:LIMit:LOAD

| | |
|-------------------|---|
| Syntax | :CALCulate:AFSPectrum:TEMPlate:LIMit:LOAD <string1> |
| Parameters | string1 is a string giving the name of the file containing the audio limits. The maximum length of string1 is 50 characters. The default for string1 is "example.lmt" . |

| | |
|--------------------|---|
| Description | This command loads the audio template description file. Please note that the data contained in the file need to be activated (using the CALC:AFSP:TEMP:LIM:STAT ON command described above) before the template will have any effect on the measurement results. |
| Query | There is no query form of this command available. |
| Example | :CALC:AFSP:LIM:LOAD "example.lmt" Loads the audio spectrum template file example.lmt which resides in the /rapid/audiolmt directory. |

:CALCulate:AFSPectrum:TEMPlate:LIMit:STORE

| | |
|--------------------|--|
| Syntax | :CALCulate:AFSPectrum:TEMPlate:LIMit:STORE <string1> |
| Parameters | string1 is a string giving the name of the file in which the current audio limits are to be saved. The maximum length of string1 is 50 characters. |
| Description | This command saves the current audio template in a file. The template can be defined by the FREQ , UPPER , LOWER and COMMENT comments. |
| Query | There is no query form of this command available. |
| Example | :CALC:AFSP:LIM:STORE "example.lmt" Stores the currently defined audio spectrum template in the file example.lmt in the /rapid/audiolmt directory. If a file of this name already exists, it will be overwritten. |

:CALCulate:AFSPectrum:TEMPlate:LIMit:COMMENT

| | |
|--------------------|---|
| Syntax | :CALCulate:AFSPectrum:TEMPlate:LIMit:COMMENT <string1> |
| Parameters | string1 is a comment line related to the coupling loss data. The maximum length of string1 is 80 characters. |
| Description | Defines a comment line for storage in the audio template file and for display in the Comment line field on the Limits screen. Note: The comment can be saved with the STORE command, together with the other template parameters. |
| Query | There is no query form of this command available. |
| Example | :CALC:AFSP:LIM:COMM "Audio template for Siemens S35" |

:CALCulate:AFSPectrum:TEMPlate:LIMit:FREQuency

| | |
|--------------------|---|
| Syntax | <code>:CALCulate:AFSPectrum:TEMPlate:LIMit:FREQuency <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>,<real9>,<real10></code> |
| 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 definitions for the upper and lower limits. |
| Query | The query form of this command is not available. |
| Example | <code>:CALC:AFSP:TEMP:LIM:FREQ 25, 50, 100, 200, 500, 1000, 2000, 4000, 8000, 16000</code> 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 | <code>:CALCulate:AFSPectrum:TEMPlate:LIMit:UPPer <real1>,<real2>,<real3>,<real4>, <real5>,<real6>,<real7>,<real8>,<real9>,<real10></code> |
| 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 | <code>:CALC:AFSP:TEMP:LIM:UPP 5, 5, 5, 5, 5, 5, 5, 5, 5, 5</code> This command sets all upper limits to +5 dB at the frequencies defined with the <code>:CALC:AFSP:TEMP:LIM:FREQ</code> 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 | <p>:CALC:RES :MEAS:ARR:PSUP:ALL :CALC:PSUP:MAV? 603.1,215.8,908.2</p> <p>Returns the average values for average power consumption (603.1 mW), average current consumption (215.8 mA) and peak current consumption (908.2 mA).</p> |

: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 | <p>:CALC:RES :MEAS:ARR:PSUP:ALL :CALC:PSUP:MMIN?</p> <p>Returns the minimum values for average power consumption (in mW), average current consumption (in mA), and peak current consumption (in mA).</p> |

: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 | <p>:CALC:RES :MEAS:ARR:PSUP:ALL :CALC:PSUP:MMAx?</p> <p>Returns the maximum values for average power consumption (in mW), average current consumption (in mA), and peak current consumption (in mA).</p> |

: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 | <p>:CALC:RES :MEAS:ARR:PSUP:ALL :CALC:PSUP:MSIG?</p> <p>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).</p> |

: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 | <p>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.</p> <p>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.</p> <p>The limits can be set using the commands described below. The results are in the sequence power consumption, average current consumption, peak current consumption.</p> |
| Example | <p>:MEAS:ARR:PSUP:CPEA 5 :CALC:PSUP:ALL:LIM?</p> <p>String returned: "1,0,0"</p> <p>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.</p> |

: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 | <p>:CALC:PSUP:ALL:LIM:STAT ON</p> <p>Switches the limit check for the current measurements on.</p> |

:CALCulate:PSUPply:ALL:LIMit:UPPer[:DATA]

| | |
|--------------------|--|
| Syntax | <code>:CALCulate:PSUPply:ALL:LIMit:UPPer[:DATA]</code> <code><real1>,<real2>,<real3></code> |
| Parameters | real1, real2 and real3 are floating point real numbers. <ul style="list-style-type: none"> - 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 | <code>:CALCulate:PSUPply:ALL:LIMit:LOWer[:DATA]</code> <code><real1>,<real2>,<real3></code> |
| Parameters | real1, real2, real3 are floating point real numbers. <ul style="list-style-type: none"> - 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 | <code>:CALCulate:PSUPply:APOWer:LIMit[:FAIL]?</code> |
| 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 | <pre> :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. </pre> |
|----------------|--|

: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 | <pre> :CALC:PSUP:APOW:LIM:STAT ON Switches the limit check of the current measurement on. </pre> |

: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 | <pre> :CALC:PSUP:APOW:LIM:UPP 100 Sets the upper limit of the power consumption measurement to 100 mW. </pre> |

: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 | <pre> :CALC:PSUP:APOW:LIM:LOW 10 Sets the lower limit of the power consumption measurement to 10 mW. </pre> |

: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 | <p>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.</p> <p>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.</p> <p>The limits can be set using the commands described below.</p> |
| Example | <pre>:CALC:PSUP:PCUR:LIM:UPP 1000.0 :CALC:PSUP:PCURrent:LIM:LOW 0 :MEAS:ARRay:PSUP:PCURrent 5 :CALC:PSUP:PCURrent:LIM?</pre> <p>String returned in this example: "1"</p> <p>This string delivered back indicates that there was a violation of the defined limits.</p> |

: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 | <p>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.</p> <p>Note All current consumption measurements require the MS Power Supply and Current Measurement options to be installed on your Willtek 4400.</p> |
| Query | There is no query form of this command available. |
| Example | <pre>:CALC:PSUP:PCUR:LIM:STAT ON</pre> <p>Switches the limit check of the peak current measurement on.</p> |

: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 | <pre>:CALC:PSUP:PCUR:LIM:UPP 1000.0</pre> <p>Sets the upper limit of the peak current consumption measurement to 1000 mA.</p> |

: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: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: <ol style="list-style-type: none"> 1. error vector magnitude rms-averaged measurement, 2. maximum error vector magnitude measurement, 3. error vector magnitude that is not exceeded 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 length, 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 | <p>realx are eight floating point real numbers.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> |
| Description | <p>Sets the upper limits for eight measurement types with one command. The order of the floating point real figures defines their meaning:</p> <p>real1 represents the upper limit of the error vector magnitude rms-averaged measurement; the physical dimension of the number stated is %.</p> <p>real2 represents the upper limit of the maximum error vector magnitude measurement; the physical dimension of the number stated is %.</p> <p>real3 represents 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 %.</p> <p>real4 represents the upper limit of the error vector magnitude origin offset measurement; the physical dimension of the number stated is dBc.</p> <p>real5 represents the upper limit of the frequency error measurement; the physical dimension of the number stated is Hertz.</p> <p>real6 represents the upper limit of the measurement of the burst length; the physical dimension of the number stated is microseconds.</p> <p>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.</p> <p>real8 represents the upper limit of the RF power level measurement; the physical dimension of the number stated is dBm.</p> |
| 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 | <p>realx are eight floating point real numbers.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> |
| Description | <p>Sets the lower limits for eight measurement types with one command. The order of the floating point real figures defines their meaning:</p> <p>real1 represents the lower limit of the error vector magnitude rms-averaged measurement; the physical dimension of the number stated is %.</p> <p>real2 represents the lower limit of the maximum error vector magnitude measurement; the physical dimension of the number stated is %.</p> <p>real3 represents 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 %.</p> <p>real4 represents the lower limit of the error vector magnitude origin offset measurement; the physical dimension of the number stated is dBc.</p> <p>real5 represents the lower limit of the frequency error measurement; the physical dimension of the number stated is Hertz.</p> <p>real6 represents the lower limit of the measurement of the burst length; the physical dimension of the number stated is microseconds.</p> <p>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.</p> <p>real8 represents the lower limit of the RF power level measurement; the physical dimension of the number stated is dBm.</p> |
| Query | There is no query form of this command available. |
| Example | <p>:CALC:EGPR:RFTX:ALL:LIM:LOW 0,0,0,-80,-90,542.8,0,0</p> <p>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.</p> |

**:CALCulate:CDMA:
RFSpectrum:ACPM:MODulation:FREQ[:DATA]**

| | |
|--------------------|--|
| Syntax | :CALCulate:CDMA: RFSpectrum:ACPM:MODulation:FREQ[:DATA] <int1>, <int1>, ... <int12> [,<PredefExp>] |
| Parameters | int1 through int12 are integer values representing frequency offsets (in kHz). There minimum value is 0 and the maximum is 2500. PredefExp is one of the following expressions: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. Each of these expressions characterizes a frequency band for which the frequencies apply. |
| Description | This command sets the 12 frequency offsets from the carrier at which adjacent channel power shall be measured. A set of offsets can be defined for each individual band. |
| Query | Returns the 12 frequencies to be sampled by the ACP measurement. A frequency band should be specified (see PredefExp above). |
| Example | :CALC:CDMA:RFSP:ACPM:MOD:FREQ:DATA 700,750,800,890, 1000,1250,1400,1600,1800,1980,2200,2400,US800 :CALC:CDMA:RFSP:ACPM:MOD:FREQ:DATA? US800 The values returned: 700,750,800,890,1000,1250,1400,1600,1800,1980,2200,2400 |

**:CALCulate:CDMA
RFSpectrum:ACPM:MODulation:LIMit[:FAIL]**

| | |
|--------------------|---|
| Syntax | :CALCulate:CDMA: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 24 boolean numbers in a string, separated by commas. These numbers indicate whether the 24 individual measurement result values of the MEAS:CDMA:RFSP:ACPM:MOD measurement did violate their individual limits (set with the commands explained above). Every boolean number represents the result of the limit check of one of the 24 measurement result values. The first two values will give the result of the limit check at the first offset (negative, then positive offset) defined with the CALC:CDMA:RFSP:ACPM:MOD:FREQ[:DATA] command. The list continues with the second pair of carrier offsets (negative and positive) and so on. |
| Example | :MEAS:CDMA:RFSP:ACPM:MOD :CALC:CDMA:RFSP:ACPM:MOD:LIM? Returns 24 numbers representing pass/results for ACP measurements at 24 frequencies. |

:CALCulate[:CDMA]:RFSP:ACPM:MOD:LIMit:STATe

| | |
|-------------------|--|
| Syntax | :CALCulate[:CDMA]:RFSP:ACPM:MOD:LIMit:STATe <PredefExp> |
| Parameters | PredefExp can take on one of the following expressions: ON OFF. Default is ON. |

| | |
|--------------------|---|
| Description | This command switches the check of the measurement result values of the MEAS:CDMA: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:CDMA:RFSP:ACPM:MOD:LIM:STAT ON Switches the limit check of the ACPM of the modulation spectrum measurement on. |

:CALC:CDMA:RFSPpectrum:ACPM:MODulation:UPPer[:DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFSPpectrum:ACPM:MODulation:UPPer[:DATA] <real1>,<real2>, ... <real12> [, <PredefExp>] |
| Parameters | real1 through real12 represent 12 floating point real numbers. The minimum value is -100.0, the maximum is 10.0, the resolution is 0.01. PredefExp is an optional predefined value used to select the band for which the upper limit values apply. Allowable values are: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. |
| Description | This command sets the 12 levels used for upper limits at the 12 frequency offsets used in the ACPM measurement. The levels are in dB. |
| Query | Returns 12 floating point values. |
| Example | :CALC:CDMA:RFSP:ACPM:MOD:UPPer:DATA -42,-42,-42,-42,-42,-42,-42,-42,-42,-42,-54,54,US800 :CALC:CDMA:RFSP:ACPM:MOD:UPPer:DATA? US800 The values returned: -42,-42,-42,-42,-42,-42,-42,-42,-42,-42,-54,-54 |

:CALC:CDMA:RFSPpectrum:ACPM:MODulation:UPPer:ENABle[:DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFSPpectrum:ACPM:MODulation:UPPer:ENABle[:DATA] <boolean1>, <boolean2>, ... <boolean12> [, <PredefExp>] |
| Parameters | boolean1 through boolean12 are 12 boolean numbers. PredefExp is an optional expression which can take on one of the following values: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800 |
| Description | This command sets the 12 boolean numbers used to indicate if the upper limits are to be applied to the adjacent channel measurements (ACPM). A 0 means that the upper limit is not applicable while a 1 indicates that it does apply. The additional parameter sets the frequency band for which the limit settings are applicable. The upper limits are set with the :CALC:CDMA:RFSPpectrum:ACPM:MODulation:UPPer[:DATA] command. |
| Query | Returns the current settings for the twelve limits. |
| Example | :CALC:CDMA:RFSP:ACPM:MOD:UPPer:ENABle:DATA 0,0,0,1,1,1,1,1,1,1,1,1,US800 :CALC:CDMA:RFSP:ACPM:MOD:UPPer:ENABle:DATA? US800 The values returned: 0,0,0,1,1,1,1,1,1,1,1,1 |

:CALC:CDMA:RFSPectrum:ACPM:MODulation:LOWer[:DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFSPectrum:ACPM:MODulation:LOWer[:DATA] <real1>,<real2>, ... <real12> [,<PredefExp>] |
| Parameters | real1 through real12 represent 12 floating point real numbers. The minimum value is -100.0, the maximum is 10.0, the resolution is 0.01. PredefExp is an optional predefined value used to select the band for which the lower limit values apply. Allowable values are: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. |
| Description | This command sets the 12 levels used for lower limits at the 12 frequency offsets used in the ACPM measurement. The levels are in dB. |
| Query | Returns 12 floating point values. |
| Example | :CALC:CDMA:RFSP:ACPM:MOD:LOWer:DATA -10,-10,-10,-60,-60,-60,-60,-80,-80,-80,-120,-120,US800 :CALC:CDMA:RFSP:ACPM:MOD:LOWer:DATA? US800 The values returned are the values set (-10, -10, -10, -60 etc.) |

:CALC:CDMA:RFSPectrum:ACPM:MODulation:LOWer:ENABLE[:DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFSPectrum:ACPM:MODulation:LOWer:ENABLE[:DATA] <boolean1>,<boolean2>, ... <boolean12> [<PredefExp>] |
| Parameters | boolean1 through boolean12 are 12 boolean numbers. PredefExp is an optional expression which can take on one of the following values: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800 |
| Description | This command sets the 12 boolean numbers used to indicate if the lower limits are to be applied to the adjacent channel measurements (ACPM). A 0 means that the lower limit is not applicable while a 1 indicates that it does apply. The additional parameter sets the frequency band for which the limit settings are applicable. The lower limits are set with the :CALC:CDMA:RFSPectrum:ACPM:MODulation:LOWer[:DATA] command. |
| Query | Returns the current settings for the twelve limits. |
| Example | :CALC:CDMA:RFSP:ACPM:MOD:LOWer:ENABLE:DATA 0,0,0,1,1,1,1,1,1,1,1,1,US800 :CALC:CDMA:RFSP:ACPM:MOD:LOWer:ENABLE:DATA? US800 The values returned: 0,0,0,1,1,1,1,1,1,1,1,1 |

:CALC:CDMA:RFSPectrum:MSpectrum:VALue

| | |
|-------------------|--|
| Syntax | :CALC:CDMA: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 <code>real1</code> parameter. <code>real1</code> 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 |
| | <ul style="list-style-type: none"> - Please note that you need to have a spectrum measurement taken before this command will return any meaningful result. We recommend to use the <code>:MEAS:CDMA:BLOC:MSP:AVG</code> command to do so. - The <code>:CALC:CDMA:RFSP:MSP:VAL</code> command is not really a measurement – it is an interpolation of the spectrum curve for the frequency specified. |
| Example | <pre>:MEAS:CDMA:BLOCdata:MSP:AVG 20 :CALC:CDMA:RFSP:MSP:VAL? -50000</pre> <p>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".</p> |

:CALC:CDMA:RFTX:FREQ:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | <code>:CALC:CDMA:RFTX:FREQ:LIMit[:FAIL]?</code> |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers boolean number in a string which indicates if the <code>:MEAS:CDMA:ARR:RFTX:FREQ</code> measurement violated its limits. |
| Example | <pre>:MEAS:CDMA:ARR:RFTX:FREQ 20 :CALC:CDMA:RFTX:FREQ:LIM?</pre> <p>The MEASurement command starts 20 frequency measurements. The query returns 1 if any of the measurement results was out of limits, or 0 if none of the results was out of limits.</p> |

:CALC:CDMA:RFTX:FREQuency:LIMit:STATe

| | |
|--------------------|---|
| Syntax | <code>:CALC:CDMA:RFTX:FREQuency:LIMit:STATe <PredefExp></code> |
| Parameters | <code>PredefExp</code> is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | <pre>:CALC:CDMA:RFTX:FREQ:LIM:STAT ON</pre> <p>Switches the limit check for the frequency error measurement on.</p> |

:CALC:CDMA:RFTX:FREQUENCY:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:FREQUENCY:LIMit:UPPer:[DATA] <int1>, <PredefExp> |
| Parameters | int1 is an integer number. The minimum value for int1 is 0, the maximum is 1000, the default is 300. PredefExp is one of the following predefined expressions: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. Default is US800. |
| Description | This command sets the upper limit of the frequency error measurement by band. The physical dimension of the number stated is Hz. |
| Query | The query form returns the stored settings. |
| Example | :CALC:CDMA:RFTX:FREQ:LIM:UPP 310, US800 :CALC:CDMA:RFTX:FREQ:LIM:UPP? US800 The query returns the previously set limit of 310 (hertz). |

:CALC:CDMA:RFTX:FREQUENCY:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:FREQUENCY:LIMit:LOWer:[DATA] <int1> <PredefExp> |
| Parameters | int1 is an integer number. The minimum value for int1 is -1000, the maximum is 0, the default is -300. PredefExp is one of the following predefined expressions: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. Default is US800. |
| Description | This command sets the lower limit of the frequency error measurement by band. The physical dimension of the number stated is Hz. |
| Query | The query form returns the stored settings. |
| Example | :CALC:CDMA:RFTX:FREQ:LIM:LOW -310, US800 :CALC:CDMA:RFTX:FREQ:LIM:LOW? US800 The query returns the previously set limit of -310 (hertz). |

:CALC:CDMA:RFTX:POWER:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:POWER:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:CDMA:ARR:RFTX:POW measurement violated the defined limits. |
| Example | :MEAS:CDMA:ARR:RFTX:POW 20 :CALC:CDMA:RFTX:POW:LIM? Returns 0 if any of the 20 power measurements was out of limits, and a 1 if all results were inside limits. |

:CALC:CDMA:RFTX:POWer:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:POWer:LIMit:STATe <PredefExp> |
| Parameters | PredefExp can take on one of the following values: ON OFF. The default is ON. |
| Description | This command switches the check of the measurement result values of the power measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:POWer:LIM:STAT ON Switches the limit check for the frequency error measurement on. |

:CALC:CDMA:RFTX:POWer:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:POWer:LIMit:UPPer:[DATA] <real>, <PredefExp1> , <PredefExp2> |
| Parameters | real is a floating point real number. Its minimum value is 0.0, the maximum is 99.0, the resolution is 0.01 and the default is 2.0. PredefExp1 can take on one of the following values: PCL1 PCL2 PCL3 PCL4 PCL5. Default is PCL1. PredefExp2 is one of the following: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. Default is US800. |
| Description | Sets the upper limit for the mobile's RF output power in a given band and power class. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:POW:LIM:UPP 2.5,PCL1,US800 :CALC:CDMA:RFTX:POW:LIM:UPP? PCL1,US800 Returns 2.5. |

:CALC:CDMA:RFTX:FREQuency:LIMit:LOWer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:FREQuency:LIMit:LOWer:[DATA] <real>, <PredefExp1> , <PredefExp2> |
| Parameters | real is a floating point real number. Its minimum value is -99.0, the maximum is 0.0, the resolution is 0.01 and the default is -99.0. PredefExp1 can take on one of the following values: PCL1 PCL2 PCL3 PCL4 PCL5. Default is PCL1. PredefExp2 is one of the following: US800 USPCs TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. Default is US800. |
| Description | Sets the lower limit for the mobile's RF output power in a given band and at a given power control level. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:FREQ:LIM:LOW -2.5,PCL1,US800 :CALC:CDMA:RFTX:FREQ:LIM:LOW? PCL1,US800 Returns -2.5. |

:CALC:CDMA:RFTX:VERRor:RMS:LIMit[:FAIL]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:VERRor:RMS:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:CDMA:ARR:RFTX:VERR:RMS measurement violated its limits. |
| Example | :MEAS:CDMA:ARR:RFTX:VERR:RMS 20 :CALC:CDMA:RFTX:VERR:RMS:LIM? Returns a 0 if any of the RMS vector error measurement results is out of limits. |

:CALC:CDMA:RFTX:VERR:RMS:LIMit:STATe

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:VERR:RMS:LIMit:STATe <PredefExp> |
| Parameters | PredefExp may take on one of the following predefined expressions: ON OFF. The default is ON. |
| Description | This command switches the check of the measurement result values of the RMS vector error measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:VERR:RMS:LIM:STAT ON Switches the limit check for the RMS-averaged vector error measurement on. |

:CALC:CDMA:RFTX:VERR:RMS:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:VERR:RMS:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 0.0, the maximum is 200.0, the resolution is 0.1, the default value is 24.4. |
| Description | Sets the upper limit for the RMS average vector error. The physical dimension of the number is percentage. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:RMS:LIM:UPP 200 :CALC:CDMA:RFTX:VERR:RMS:LIM:UPP? The query returns the previously set value 200.0. |

:CALC:CDMA:RFTX:VERR:RMS:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:RMS:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 0.0, the maximum is 200.0, the resolution is 0.1, the default value is 0.0. |
| Description | Sets the lower limit for the RMS average vector error. The physical dimension of the number stated is percentage. |
| Query | There is a query form which returns the stored settings. |

| | |
|----------------|---|
| Example | <pre>:CALC:CDMA:RFTX:VERR:RMS:LIM:LOW 200 :CALC:CDMA:RFTX:VERR:RMS:LIM:LOW? The query returns the previously set value 200.0.</pre> |
|----------------|---|

:CALC:CDMA:RFTX:VError:PEAK:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VError:PEAK:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:CDMA:ARR:RFTX:VError:PEAK measurement violated its limits. |
| Example | <pre>:MEAS:CDMA:ARR:RFTX:VERR:PEAK 20 :CALC:CDMA:RFTX:VERR:PEAK:LIM? Returns 0 if all 20 results are within limits, or 1 if any result is outside the limits.</pre> |

:CALC:CDMA:RFTX:VError:PEAK:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VError:PEAK:LIMit:STATe <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | <pre>:CALC:CDMA:RFTX:VERR:PEAK:LIM:STAT ON Switches the limit check for the peak vector error measurement on.</pre> |

:CALC:CDMA:RFTX:VErr:PEAK:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VErr:PEAK:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is 0.0, the maximum is 200.0, the resolution is 0.1, and the default value is 200.0. |
| Description | Sets the upper limit for the peak vector error measurement. The physical dimension of the number is percentage. |
| Query | There is query form which returns the stored settings. |
| Example | <pre>:CALC:CDMA:RFTX:VERR:PEAK:LIM:UPP 200 :CALC:CDMA:RFTX:VERR:PEAK:LIM:UPP? Returns 200.</pre> |

:CALC:CDMA:RFTX:VErr:PEAK:LIMit:LOWer:[DATA]

| | |
|-------------------|---|
| Syntax | :CALC:CDMA:RFTX:VErr:PEAK:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is 0.0, the maximum is 200.0, the resolution is 0.1, the default value is 0.0. |

| | |
|--------------------|---|
| Description | Sets the upper limit. The physical dimension of the number is percentage. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:PEAK:LIM:LOWer 0 :CALC:CDMA:RFTX:VERR:PEAK:LIM:LOWer? Returns 0. |

:CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit[:FAIL]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the RMS phase error measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:VERR:PHAS:RMS? 5 :CALC:CDMA:RFTX:PHASe:RMS:LIM? Returns 1 if any of the five results exceeds the given limits. |

:CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe: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 RMS phase error measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:STAT ON Switches the limit check for the RMS phase error measurement on. |

:CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit:UPPer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value is 0.0, the maximum is 45, the resolution is 0.01, the default value is 14. |
| Description | Sets the upper limit for the peak vector error measurement. The physical dimension of the number stated is degrees. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:UPP 45 :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:UPP? The returned result is 45.0. |

:CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is 0, the maximum is 45, the resolution is 0.01, the default value is 0. |
| Description | Sets the upper limit of the peak vector error measurement. The physical dimension of the number stated is degrees. |
| Query | There is query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:LOWer -44.0 :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:LOW? The query returns 44.0. |

:CALC:CDMA:RFTX:VERR:PHASe:PEAK:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:PEAK:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the peak phase measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:VERR:PHAS:PEAK? 5 :CALC:CDMA:RFTX:VERR:PHASe:PEAK:LIM? Returns 1 if any of the results exceeded the limits. |

:CALC:CDMA:RFTX:VERR:PEAK:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PEAK:LIMit:STATe <PredefinExp> |
| Parameters | PredefinExp 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 peak vector error measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:VERR:PEAK:LIM:STAT ON Switches the limit check for the peak phase measurement on. |

:CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is -45.0, the maximum is 45.0, the resolution is 0.01, the default value is 45.0. |
| Description | Sets the upper limit of the RMS phase error. The physical dimension of the number stated is degrees. |

| | |
|----------------|--|
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:UPP 45 :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:UPP? Returns the upper limit of 45.0 (degrees). |

:CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit:LOWer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is -45.0, the maximum is 45.0, the resolution is 0.01, the default value is 45.0. |
| Description | Sets the upper limit of the RMS phase error measurement. The physical dimension of the number stated is degrees. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:LOW 45 :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:LOW? Returns -45.0. |

**:CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:RMS
:LIMit[:FAIL]**

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:RMS:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the RMS phase magnitude measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:VERR:PHAS:MAGNitude:RMS? 10 :CALC:CDMA:RFTX:VERR:PHASe:MAGN:RMS:LIM? Returns 0 if all the results are within the limits. |

**:CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:RMS:LIMit
:STATe**

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude: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 result values of the RMS phase magnitude measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:MAGN:RMS:LIM:STAT ON Switches the limit check for the RMS phase magnitude measurement on. |

**:CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:RMS:LIMit
:UPPer:[DATA]**

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:RMS:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is 0, the maximum is 200, the resolution is 0.1, the default value is 24.4. |
| Description | Sets the upper limit of the RMS phase magnitude measurement. The physical dimension of the number stated is percent. |
| Query | There is query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:RMS:LIM:UPP 200.0 :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:UPP? Returns the setting of 200.0. |

**:CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:RMS:LIMit
:LOWer:[DATA]**

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:RMS:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 0.0, the maximum is 200.0, the resolution is 0.1, the default value is 0.0. |
| Description | This command sets the lower limit of the RMS phase magnitude. The physical dimension of the number stated is percent. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:RMS:LIM:LOW 0.0 :CALC:CDMA:RFTX:VERR:PHASe:RMS:LIM:UPP? Returns 0.0. |

**:CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK
:LIMit[:FAIL]**

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the peak phase magnitude measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:PHAS:MAGNitude:PEAK? 5 :CALC:CDMA:RFTX:PHASe:MAGN:PEAK:LIM? Returns 0 if none of the results exceeded the limits. |

**:CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK:LIMit
:STATe**

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK:LIMit:STATe <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:MAGN:PEAK:LIM:STAT ON Switches the limit check for the peak magnitude measurement on. |

**:CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK:LIMit
:UPPer:[DATA]**

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK:LIMit:UPPer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value for parameter is 0.0, the maximum is 200.0, the resolution is 0.1, the default value is 200.0. |
| Description | Sets the upper limit for the peak phase magnitude. The physical dimension of the number stated is percent. |
| Query | There is query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK:LIM:UPP 200.0 :CALC:CDMA:RFTX:VERR:PHASe:PEAK:LIM:UPP? Returns 200.0. |

**:CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK:LIMit
:LOWer:[DATA]**

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK:LIMit:LOWer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value for parameter is 0.0, the maximum is 200.0, the resolution is 0.1, the default value is 0.0. |
| Description | Sets the lower limit of the peak phase magnitude. The physical dimension of the number stated is percent. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:VERR:PHASe:MAGNitude:PEAK:LIM:LOW 0.0 :CALC:CDMA:RFTX:VERR:PHASe:PEAK:LIM:LOW? Returns 0.0. |

:CALC:CDMA:RFTX:TIMEoffset:LIMit[:FAIL]

| | |
|-------------------|--|
| Syntax | :CALC:CDMA:RFTX:TIMEoffset:LIMit[:FAIL]? |
| Parameters | There are no parameters. |

| | |
|--------------------|---|
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the time offset error measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:TIMEoffset? 5 :CALC:CDMA:RFTX:TIM:LIM? Returns 1 if the limits are exceeded in any of the time offset error measurements. |

:CALC:CDMA:RFTX:TIMEoffset:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:TIMEoffset: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 time offset error measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:TIMEoffset:LIM:STAT ON Switches the limit check for the time offset measurement on. |

:CALC:CDMA:RFTX:TIMEoffset:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:TIMEoffset:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is -5, the maximum is 5, the resolution is 0.1, the default value is 1. |
| Description | Set this sets upper limit. The physical dimension of the number stated is nanoseconds. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:TIMEoffset:LIM:UPP 5.0 :CALC:CDMA:RFTX:TIM:LIM:UPP? Value returned is "5.0". |

:CALC:CDMA:RFTX:TIMEoffset:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:TIMEoffset:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is -5, the maximum is 5, the resolution is 0.1, the default value is -1. |
| Description | Sets the lower limit for the time offset error measurement. The physical dimension of the number stated is nanoseconds. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:TIMEoffset:LIM:LOW -5.0 :CALC:CDMA:RFTX:TIM:LIM:LOW? Value returned is -5.0. |

:CALC:CDMA:RFTX:WQUality:LIMit[:FAIL]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:WQUality:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the waveform quality measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:WQUality? 1 :CALC:CDMA:RFTX:WQU:LIM? Returns 0 if the result is within limits. |

:CALC:CDMA:RFTX:WQUality:LIMit:STATe

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:WQUality:LIMit:STATe <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the waveform quality measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:WQUality:LIM:STAT ON Switches the limit check for the waveform quality measurement on. |

:CALC:CDMA:RFTX:WQUality:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:WQUality:LIMit:UPPer:[DATA] <PredefExp> |
| Parameters | PredefExp is a floating point real number. The minimum value for this parameter is 0.90, the maximum is 1.0, the resolution is 0.0001, the default value is 1.0. |
| Description | Sets the upper limit for the waveform quality measurement. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:WQUality:LIM:UPP 1.0 :CALC:CDMA:RFTX:WQU:LIM:UPP? Returns 1.0. |

:CALC:CDMA:RFTX:WQUality:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:WQUality:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for this parameter is 0.9, the maximum is 1.0, the resolution is 0.0001, the default value is 0.944. |
| Description | Sets the lower limit for the waveform quality measurement. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:WQUality:LIM:LOW 0.944 :CALC:CDMA:RFTX:WQU:LIM:LOW? Returns 0.944. |

:CALC:CDMA:RFTX:STANdby:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:STANdby:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the standby power measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:STANdby? 1 :CALC:CDMA:RFTX:STAN:LIM? Returns 1 if the result exceeds the limits. |

:CALC:CDMA:RFTX:STANdby:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:STANdby:LIMit:STATe <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:STAN:LIM:STAT ON Switches the limit check for the standby power measurement on. |

:CALC:CDMA:RFTX:STANdby:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:STANdby:LIMit:UPPer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value for this parameter is -200, the maximum is 36, the resolution is 0.01, the default value is -61. |
| Description | Sets the upper limit for the standby power measurement. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:STANdby:LIM:UPP 1.0 :CALC:CDMA:RFTX:STAN:LIM:UPP? Returns 1.0. |

:CALC:CDMA:RFTX:STANdby:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:STANdby:LIMit:LOWer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value for parameter is -70.0, the maximum is 36.0, the resolution is 0.01, the default value is -70.0. |
| Description | Sets the lower limit for the standby power measurement. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |

| | |
|----------------|--|
| Example | :CALC:CDMA:RFTX:STANdby:LIM:LOWer -70.0 :CALC:CDMA:RFTX:STAN:LIM:LOW? Returned value is -70.0. |
|----------------|--|

:CALC:CDMA:RFTX:MINimum:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:MINimum:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the minimum power measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:MINimum? 5 :CALC:CDMA:RFTX:MIN:LIM? Returns 1 if any of the results exceeds the limits. |

:CALC:CDMA:RFTX:MINimum:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:MINimum:LIMit:STATe <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:MINimum:LIM:STAT ON Switches the limit check for the minimum power measurement on. |

:CALC:CDMA:RFTX:MINimum:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:MINimum:LIMit:UPPer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value for parameter is -200, the maximum is 36, the resolution is 0.1, the default value is -50. |
| Description | Sets the upper limit for the minimum power measurement. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:MINimum:LIM:UPP 36.0 :CALC:CDMA:RFTX:MIN:LIM:UPP? Returns 36.0. |

:CALC:CDMA:RFTX:MINimum:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:MINimum:LIMit:LOWer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value for parameter is -200, the maximum is 36, the resolution is 0.1, the default value is -70. |
| Description | Set this sets lower limit. The physical dimension of the number stated is dBm. |

| | |
|----------------|--|
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:MINimum:LIM:LOWer -70.0 :CALC:CDMA:RFTX:MIN:LIM:LOW? Returns -70.0. |

:CALC:CDMA:RFTX:ACCesschannel:LIMit[:FAIL]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:ACCesschannel:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the access channel power measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:ACCesschannel? 1 :CALC:CDMA:RFTX:ACC:LIM? Returns 1 if the result exceeds the limits. |

:CALC:CDMA:RFTX:ACCesschannel:LIMit:STATE

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:ACCesschannel:LIMit:STATE <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:ACC:LIM:STAT ON Switches the limit check for the access channel power measurement on. |

:CALC:CDMA:RFTX:Accesschannel:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:Accesschannel:LIMit:UPPer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value for this parameter is -70.0, the maximum is 36, the resolution is 0.1, the default value is 9.5. |
| Description | Sets the upper limit for the access channel power measurement. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:ACC:LIM:UPP 36.0 :CALC:CDMA:RFTX:ACC:LIM:UPP? Returns 36.0. |

:CALC:CDMA:RFTX:ACCesschannel:LIMit:LOWer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:ACCesschannel:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for this parameter is -70, the maximum is 36, the resolution is 0.1, the default value is -9.5. |
| Description | Sets the upper limit for the access channel power measurement. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:ACC:LIM:LOWer -70.0 :CALC:CDMA:RFTX:ACC:LIM:LOW? Returns -70.0. |

:CALC:CDMA:RFTX:CLOSeDloop:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CLOSeDloop:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the closed loop power measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:CLOSeDloop? 1 :CALC:CDMA:RFTX:CLOS:LIM? |

:CALC:CDMA:RFTX:CLOSeDloop:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CLOSeDloop:LIMit:STATe <PreDefExp> |
| Parameters | PreDefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:CLOS:LIM:STAT ON Switches the limit check for the closed-loop power measurement on. |

:CALC:CDMA:RFTX:CLOSeDloop:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:CLOSeDloop:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 0, the maximum is 99, the resolution is 0.01, the default value is 24. |
| Description | Sets the upper limit for the closed loop power measurement. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |

Example :CALC:CDMA:RFTX:CLOSeDloop:LIM:UPP 36.0
:CALC:CDMA:RFTX:CLOS:LIM:UPP? Returns 36.0.

:CALC:CDMA:RFTX:CLOSeDloop:LIMit:LOWer:[DATA]

Syntax :CALC:CDMA:RFTX:CLOSeDloop:LIMit:LOWer:[DATA] <Real1>
Parameters Real1 is a floating point real number. The minimum value is -99, the maximum is 0, the resolution is 0.1, the default value is -24.
Description Sets the lower limit for the closed loop power measurement. The physical dimension of the number stated is dBm.
Query There is a query form which returns the stored settings.
Example :CALC:CDMA:RFTX:CLOSeDloop:LIM:LOWer -70.0
:CALC:CDMA:RFTX:CLOS:LIM:LOW? Returns -70.0.

:CALC:CDMA:RFTX:CCHNoise:LIMit[:FAIL]

Syntax :CALC:CDMA:RFTX:CCHNoise:LIMit[:FAIL]?
Parameters There are no parameters.
Description There is solely a query form of this command available.
Query This command delivers a boolean number in a string which indicates if the code channel noise measurement exceeds the limits.
Example :MEAS:CDMA:ARR:RFTX:CCHNoise? 10
:CALC:CDMA:RFTX:CCHN:LIM?

:CALC:CDMA:RFTX:CCHNoise:LIMit:STATe

Syntax :CALC:CDMA:RFTX:CCHNoise: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 code channel noise measurement against their user-definable limits either on or off.
Query There is no query form of this command available.
Example :CALC:CDMA:RFTX:CCHNoise:LIM:STAT ON
Switches the limit check for the code channel noise measurement on.

:CALC:CDMA:RFTX:CCHNoise:LIMit:UPPer:[DATA]

Syntax :CALC:CDMA:RFTX:CCHNoise:LIMit:UPPer:[DATA] <Real1>
Parameters Real1 is a floating point real number. The minimum value for parameter is -99, the maximum is 0, the resolution is 0.1, the default value is -23.
Description Sets the upper limit for the code channel noise measurement. The physical dimension of the number stated is dBm.

| | |
|----------------|--|
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:CCHNoise:LIM:UPP -25.0 :CALC:CDMA:RFTX:CCHN:LIM:UPP? Value returned: -25.0. |

:CALC:CDMA:RFTX:CCHNoise:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CCHNoise:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is -40.0, the maximum is 0.0, the resolution is 0.1, the default value is -40.0. |
| Description | Sets the lower limit for the code channel noise measurement. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:CCHNoise:LIM:LOW -40.0 :CALC:CDMA:RFTX:CCHN:LIM:LOW? Value returned: -40.0. |

:CALC:CDMA:RFTX:CDPower:LIMit[:FAIL]?

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:CDPower:LIMit[:FAIL]? <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: RPICH RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. Default is RPICH. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the code domain power measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:CDPower? 1,ALL :CALC:CDMA:RFTX:CDPO:LIM? RPICH Returns 0 if the code domain power measurement on the RPICH is within limits. |

:CALC:CDMA:RFTX:CDPower:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CDPower: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 code domain power measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:CDPO:LIM:STAT ON Switches the limit check for the code domain power measurement on. |

:CALC:CDMA:RFTX:CDPower:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CDPower:LIMit:UPPer:[DATA] <Real1> <PredefExp> |
| Parameters | Real1 is a floating point real number. The minimum value for this parameter is -40.0, the maximum is 0.0, the resolution is 0.1, the default value is 0.0. PredefExp is one of the following predefined expressions: RPICH RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. Default is RPICH. |
| Description | Sets the upper limit for the code domain power measurement. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:CDPower:LIM:UPP 0.0 :CALC:CDMA:RFTX:CDP:LIM:UPP? Returns 0.0. |

:CALC:CDMA:RFTX:CDPower:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CDPower:LIMit:LOWer:[DATA] <Real1> <PredefExp> |
| Parameters | Real1 is a floating point real number. The minimum value for this parameter is -40.0, the maximum is 0.0, the resolution is 0.1, the default value is 0.0. PredefExp is one of the following predefined expressions: RPICH RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. Default is RPICH. |
| Description | Sets the lower limit for the code domain power measurement. The physical dimension of the number stated is dBm. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:CDPower:LIM:LOW -40.0 :CALC:CDMA:RFTX:CDP:LIM:LOW? Returns -40.0. |

:CALC:CDMA:RFTX:CDPhase:LIMit[:FAIL]?

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CDPhase:LIMit[:FAIL]? <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: RPICH RDCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. Default is RPICH. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the code domain phase measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:CDPhase? 10,ALL :CALC:CDMA:RFTX:CDPH:LIM? RPICH Returns 1 if any of the results fails the limits. |

:CALC:CDMA:RFTX:CDPhase:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CDPhase: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 code domain phase measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:CDPhase:LIM:STAT ON Switches the limit check for the code domain phase measurement on. |

:CALC:CDMA:RFTX:CDPhase:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:CDPhase:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 0.0, the maximum is 0.5, the resolution is 0.0001, the default value is 0.5. |
| Description | Sets the upper limit for the code domain phase measurement. The physical dimension of the number stated is radians. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:CDPhase:LIM:UPP 0.5 :CALC:CDMA:RFTX:CDPhase:LIM:UPP? Returns 0.5. |

:CALC:CDMA:RFTX:CDPhase:LIMit:LOWer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:CDPhase:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 0.0, the maximum is 0.5, the resolution is 0.0001, the default value is 0. |
| Description | Sets the lower limit for the code domain phase measurement. The physical dimension of the number stated is radians. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:CDPhase:LIM:LOW 0.1 :CALC:CDMA:RFTX:CDPhase:LIM:LOW? Returns 0.1. |

:CALC:CDMA:RFTX:CDTime:LIMit[:FAIL]?

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:CDTime:LIMit[:FAIL]? <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: RPICH RDCCCh RSCH2x2 RSCH2x4 RFCH RCCCh RSCH1x4 RSCH1x8. Default is RPICH. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the code domain time measurement exceeds the limits. |

| | |
|----------------|---|
| Example | :MEAS:CDMA:ARR:RFTX:CDTime? 1,ALL :CALC:CDMA:RFTX:CDT:LIM? RPICH |
|----------------|---|

:CALC:CDMA:RFTX:CDTime:LIMit:STATe

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:CDTime: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 code domain time measurement against their user-definable limits either on or off. |
| Query | There is no query form of this command available. |
| Example | :CALC:CDMA:RFTX:CDTime:LIM:STAT ON Switches the limit check for the code domain time measurement on. |

:CALC:CDMA:RFTX:CDTime:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CDTime:LIMit:UPPer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value is 0.0, the maximum is 200.0, the resolution is 0.5, the default value is 200.0. |
| Description | Sets the upper limit for the code domain time measurement. The physical dimension of the number is stated as a percentage. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:CDTime:LIM:UPP 200.0 :CALC:CDMA:RFTX:CDT:LIM:UPP? Value returned is 200.0. |

:CALC:CDMA:RFTX:CDTime:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:CDTime:LIMit:LOWer:[DATA] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value is 0.0, the maximum is 200.0, the resolution is 0.5, the default value is 0. |
| Description | Sets the lower limit for the code domain time measurement. The physical dimension of the number is stated as a percentage. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:CDTime:LIM:LOW 10 :CALC:CDMA:RFTX:CDT:LIM:LOW? Value returned is 10.0. |

:CALC:CDMA:RFTX:GATedptt:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:GATedptt:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |

| | |
|----------------|---|
| Query | This command delivers a boolean number in a string which indicates if the gated power/time template measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:GATedptt? 1 :CALC:CDMA:RFTX:GATedptt:LIM? |

:CALC:CDMA:RFTX:GATedptt:LIMit:UPPer1:USEC:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:GATedptt:LIMit:UPPer1:USEC:[DATA] <Int1>, &PredefExp> :CALC:CDMA:RFTX:GATedppt:LIM:UPPer2:USEC:[DATA] <Int1>, &PredefExp> |
| Parameters | Int1 is an integer number. The minimum value for Int1 is 0, the maximum is 2500, the default value is 0. PredefExp is one of the following predefined expressions: RC12 RC34. Default is RC12. |
| Description | Sets the upper limit for the gated power/time template measurement. The physical dimension of the number stated is microseconds. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:GATedptt:LIM:UPP1:USEC 0,RC12 :CALC:CDMA:RFTX:GAT:LIM:UPP1:USEC? Returned string: 0,RC12 :CALC:CDMA:RFTX:GATedppt:LIM:UPP2:USEC 1250,RC12 :CALC:CDMA:RFTX:GAT:LIM:UPP2:USEC? Returns 1250,RC12 |

:CALC:CDMA:RFTX:GATedptt:LIMit:UPPer1:DB:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:GATedptt:LIMit:UPPer1:DB:[DATA] <Real1>, <PredefExp> :CALC:CDMA:RFTX:GATedppt:LIM:UPPer2:DB:[DATA] <Real1> <PredefExp> |
| Parameters | Real1 is a floating point real number. The minimum value for Real1 is -70, the maximum is 10, the resolution is 0.1, the default value is -6.0. PredefExp is one of the following predefined expressions: RC12 RC34. Default is RC12. |
| Description | Sets the upper limit for the gated power/time template measurement. The physical dimension of the number stated is dB. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:GATedptt:LIM:UPP1:DB 0.0,RC12 :CALC:CDMA:RFTX:GAT:LIM:UPP1:DB? Returned string: 0.0,RC12 :CALC:CDMA:RFTX:GATedppt:LIM:UPP2:DB 10.0,RC12 :CALC:CDMA:RFTX:GAT:LIM:UPP2:DB? Returned string: 10.0,RC12. |

:CALC:CDMA:RFTX:GATedptt:LIMit:LOWer1:USEC:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:GATedptt:LIMit:LOWer1:USEC:[DATA] <Int1>, &PredefExp :CALC:CDMA:RFTX:GATedptt:LIM:LOWer2:USEC:[DATA] <Int1>, &PredefExp |
| Parameters | Int1 is an integer number. The minimum value for Int1 is 0, the maximum is 2500, the default value is 0. PredefExp is one of the following predefined expressions: RC12 RC34. Default is RC12. |
| Description | Sets the upper limit for the gated power/time template measurement. The physical dimension of the number stated is microseconds. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:GATedptt:LIM:LOW1:USEC 0,RC12 :CALC:CDMA:RFTX:GAT:LIM:LOW1:USEC? Returned string: 0,RC12 :CALC:CDMA:RFTX:GATedptt:LIM:LOW2:USEC 1250,RC12 :CALC:CDMA:RFTX:GAT:LIM:LOW2:USEC? Returns 1250,RC12 |

:CALC:CDMA:RFTX:GATedptt:LIMit:LOWer1:DB:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:GATedptt:LIMit:LOWer1:DB:[DATA] <Real1>, <PredefExp> :CALC:CDMA:RFTX:GATedptt:LIM:LOWer2:DB:[DATA] <Real1> <PredefExp> |
| Parameters | Real1 is a floating point real number. The minimum value for Real1 is -90, the maximum is 10, the resolution is 0.1, the default value is 0.0. PredefExp is one of the following predefined expressions: RC12 RC34. Default is RC12. |
| Description | Sets the upper limit for the gated power/time template measurement. The physical dimension of the number stated is dB. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:GATedptt:LIM:UPP1:DB 0.0,RC12 :CALC:CDMA:RFTX:GAT:LIM:UPP1:DB? Returned string: 0.0,RC12 :CALC:CDMA:RFTX:GATedptt:LIM:UPP2:DB 10.0,RC12 :CALC:CDMA:RFTX:GAT:LIM:UPP2:DB? Returned string: 10.0,RC12. |

:CALC:CDMA:RFTX:GATedptt:LIMit:AVERages:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:GATedptt:LIMit:AVERages:[DATA] <Int1>, <PredefExp> :CALC:CDMA:RFTX:GATedptt:LIM:AVER:[DATA] <Int1>, <PredefExp> |
| Parameters | Int1 is an integer number. The minimum value for Int1 is 0, the maximum is 200, the default value is 100. PredefExp is one of the following predefined expressions: RC12 RC34. Default is RC12. |
| Description | Sets the number of gates to average. |

| | |
|----------------|---|
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:GATedptt:LIM:AVERage 50,RC12 :CALC:CDMA:RFTX:GAT:LIM:AVER? Returns 50,RC12. :CALC:CDMA:RFTX:GATedppt:LIM:AVERage 100,RC34 :CALC:CDMA:RFTX:GAT:LIM:AVER? Returned string is 100,RC34. |

:CALC:CDMA:RFTX:OPENloop:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:OPENloop:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the open loop power control measurement exceeds the limits. |
| Example | :MEAS:CDMA:ARR:RFTX:OPENloop? 1 :CALC:CDMA:RFTX:OPEN:LIM? |

:CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:MAXium

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:MAXium <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is 100, the maximum is 200, the resolution is 1.0, the default value is 120. |
| Description | Sets the upper limit for the open loop power control measurement. The physical dimension of the number stated is percent. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:MAXimum 120 :CALC:CDMA:RFTX:OPEN:LIM:UPP:MAX? Returned string: 120. |

:CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:OFFset

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:OFFset <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is 0.0, the maximum is 10, the resolution is 0.1, the default value is 1.5. |
| Description | Sets the upper limit for the offset in the open loop power control measurement. The physical dimension of the number stated is dB. |
| Query | There is query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:OFFSet 1.5 :CALC:CDMA:RFTX:OPEN:LIM:UPP:OFFSet? Returned string: 1.5. |

:CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:VARA

| | |
|--------------------|---|
| Syntax | :CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:VARA <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is 0.0, the maximum is 5, the resolution is 0.1, the default value is 0.5. |
| Description | Sets variable A when (t>=24) value. This variable is used to calculate the upper limit. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:VARA 0.5 :CALC:CDMA:RFTX:OPEN:LIM:UPP:VARA? Returned string: 0.5. |

:CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:VARB

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:VARB <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is 0, the maximum is 10, the resolution is 0.1, the default value is 2. |
| Description | Sets variable B when t<24, used to calculate the upper limit. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:OPENloop:LIMit:UPPer:VARB 2.0 :CALC:CDMA:RFTX:OPEN:LIM:UPP:VARB? Returns 2.0. |

:CALC:CDMA:RFTX:OPENloop:LIMit:LOWer:MINimum

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:OPENloop:LIMit:LOWer:MINimum <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 1, the maximum is 100, the resolution is 1.0, the default value is 80. |
| Description | Sets the lower limit for the open loop power control measurement. The physical dimension of the number stated is percent. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:OPENloop:LIMit:LOWer:MINimum 80 :CALC:CDMA:RFTX:OPEN:LIM:UPP:MAX? Returns 80 |

:CALC:CDMA:RFTX:OPENloop:LIMit:LOWer:OFFSet

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:OPENloop:LIMit:LOWer:OFFSet <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is -10.0, the maximum is 0, the resolution is 0.1, the default value is -1.0. |
| Description | Sets the lower limit for the offset in the open loop power control measurement. The physical dimension of the number stated is dB. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:CDMA:RFTX:OPENloop:LIMit:LOWer:OFFSet -1.0 :CALC:CDMA:RFTX:OPEN:LIM:LOW:OFFSet? Returns -1.0 |

:CALC:CDMA:RFTX:OPENloop:LIMit:LOWer:VARC

| | |
|--------------------|--|
| Syntax | :CALC:CDMA:RFTX:OPENloop:LIMit:LOWer:VARC |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:AMPS:ARR:RFTX:POW measurement violated the defined limits. |
| Example | :CALC:CDMA:RFTX:OPENloop:LIMit:LOWer:VARC 2.0 :CALC:CDMA:RFTX:OPEN:LIM:LOW:VARC? Returns 2.0 |

:CALC:AMPS:RFTX:POWER:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:POWER:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:AMPS:ARR:RFTX:POW measurement violated the defined limits. |
| Example | :MEAS:AMPS:ARR:RFTX:POW 20 :CALC:AMPS:RFTX:POWER:LIM? Returns 0 if any of the 20 power measurements was out of limits, and 1 if all results were inside limits. |

:CALC:AMPS:RFTX:POWER:LIMit:STATE

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:POWER:LIMit:STATE <PredefExp> |
| Parameters | PredefExp can take on one of the following values: ON OFF. The default is ON. |
| Description | This command switches the check of the measurement result values of the power measurement against their user-definable limits either ON or OFF. |
| Query | There is no query form of this command available. |
| Example | :CALC:AMPS:RFTX:POWER:LIM:STAT ON Switches the limit check for the power measurement on. |

:CALC:AMPS:RFTX:POWER:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:POWER:LIMit:UPPer:[DATA] <real>, <int1> |
| Parameters | real is a floating point real number. Its minimum value is 0.0, the maximum is 99.0, the resolution is 0.01 and the default is 2.0. int1 specifies the MAC level which the limit will apply to. The range is 0 to 7. Default is 0. |
| Description | Sets the upper limit for the mobile's RF output power at a given MAC lev. The physical dimension of the number stated is dB. |
| Query | There is a query form which returns the stored settings. |

| | |
|----------------|---|
| Example | :CALC:AMPS:RFTX:POW:LIM:UPP 2.5,1 :CALC:AMPS:RFTX:POW:LIM:UPP? 1 Returns 2.5. |
|----------------|---|

:CALC:AMPS:RFTX:FREQuency:LIMit:LOWer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFTX:FREQuency:LIMit:LOWer:[DATA] <real>,<int1> |
| Parameters | real is a floating point real number. Its minimum value is -99.0, the maximum is 0.0, the resolution is 0.01 and the default is -99.0. PredefExp1 can take on one of the following values: PCL1 PCL2 PCL3 PCL4 PCL5. Default is PCL1. int1 specifies the MAC level which the limit will apply to. The range is 0 to 7. Default is 0. |
| Description | Sets the lower limit for the mobile's RF output power at a given power MAC level. The physical dimension of the number stated is dB. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:MAC:LIM:LOW -2.5,1 :CALC:AMPS:RFTX:MAC:LIM:LOW? 1 Returns -2.5. |

:CALC:AMPS:RFTX:DEV:SAT:LIMit[:FAIL]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFTX:DEV:SAT:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:AMPS:ARR:RFTX:SAT:DEV measurement violated its limits. |
| Example | :MEAS:AMPS:ARR:RFTX:DEV:SAT 20 CALC:AMPS:RFTX:DEV:SAT:LIM? Returns 0 if any of the SAT DEVIation measurement results is out of limits. |

:CALC:AMPS:RFTX:DEV:SAT:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:DEV:SAT:LIMit:STATe <PredefExp> |
| Parameters | PredefExp may take on one of the following predefined expressions: ON OFF. The default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either ON or OFF. |
| Query | There is no query form of this command available. |
| Example | :CALC:AMPS:RFTX:DEV:SAT:LIM:STAT ON Switches the limit check for the SAT deviation measurement on. |

:CALC:AMPS:RFTX:DEV:SAT:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:DEV:SAT:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 1600, the maximum is 2400, the resolution is 1, the default is value is 2200. |
| Description | Sets the upper limit for the SAT Deviation measurement. The physical dimension of the number stated is Hz. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:DEV:SAT:LIM:UPP 2200 :CALC:AMPS:RFTX:DEV:SAT:LIM:UPP? The query returns the previously set value 2200. |

:CALC:AMPS:RFTX:DEV:SAT:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:DEV:SAT:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 1600, the maximum is 2400, the resolution is 1, the default is value is 1800. |
| Description | Sets the lower limit for the SAT Deviation measurement. The physical dimension of the number stated is Hz. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:DEV:SAT:LIM:LOW 1800 :CALC:AMPS:RFTX:DEV:SAT:LIM:LOW? The query returns the previously set value 1800. |

:CALC:AMPS:RFTX:DEV:ST:LIMit[:FAIL]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFTX:DEV:ST:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:AMPS:ARR:RFTX:ST:DEV measurement violated its limits. |
| Example | :MEAS:AMPS:ARR:RFTX:DEV:ST 20 :CALC:AMPS:RFTX:DEV:ST:LIM? Returns a 0 if any of the ST DEVIation measurement results is out of limits. |

:CALC:AMPS:RFTX:DEV:ST:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:DEV:ST:LIMit:STATe <PredefExp> |
| Parameters | PredefExp may take on one of the following predefined expressions: ON OFF. The default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either ON or OFF. |

| | |
|----------------|---|
| Query | There is no query form of this command available. |
| Example | :CALC:AMPS:RFTX:DEV:ST:LIM:STAT ON Switches the limit check for the ST deviation measurement on. |

:CALC:AMPS:RFTX:DEV:ST:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:DEV:ST:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 6400, the maximum is 9600, the resolution is 1, the default is value is 8800. |
| Description | Sets the upper limit for the ST Deviation measurement. The physical dimension of the number stated is Hz. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:DEV:ST:LIM:UPP 8800 :CALC:AMPS:RFTX:DEV:ST:LIM:UPP? The query returns the previously set value 8800. |

:CALC:AMPS:RFTX:DEV:ST:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:DEV:ST:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 6400, the maximum is 9600, the resolution is 1, the default is value is 7200. |
| Description | Sets the lower limit for the ST Deviation measurement. The physical dimension of the number stated is Hz. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:DEV:ST:LIM:LOW 7200 :CALC:AMPS:RFTX:DEV:ST:LIM:LOW? The query returns the previously set value 7200. |

:CALC:AMPS:RFTX:DEV:AUD:LIMit[:FAIL]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFTX:DEV:AUD:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:AMPS:ARR:RFTX:DEV:AUD measurement violated its limits. |
| Example | :MEAS:AMPS:ARR:RFTX:DEV:AUD 20 :CALC:AMPS:RFTX:DEV:AUD:LIM? Returns 0 if all 20 results are within limits, or 1 if any result is outside the limits. |

:CALC:AMPS:RFTX:DEV:AUD:LIMit:STATe

| | |
|---------------|---|
| Syntax | :CALC:AMPS:RFTX:DEV:AUD:LIMit:STATe <PredefExp> |
|---------------|---|

| | |
|--------------------|---|
| Parameters | PreDefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either ON or OFF. |
| Query | There is no query form of this command available. |
| Example | :CALC:AMPS:RFTX:DEV:AUD:LIM:STAT ON Switches the limit check for the audio deviation measurement on. |

:CALC:AMPS:RFTX:DEV:AUD:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFTX:DEV:AUD:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is 0, the maximum is 30000, the resolution is 1, and the default is value is 12000. |
| Description | Sets the upper limit for the audio deviation measurement. The physical dimension of the number is stated as a Hz. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:DEV:AUD:LIM:UPP 12000 :CALC:AMPS:RFTX:DEV:AUD:LIM:UPP? Returns 12000. |

:CALC:AMPS:RFTX:DEV:AUD:LIMit:LOWer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFTX:DEV:AUD:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is 0.0, the maximum is 200.0, the resolution is 0.1, and the default is value is 0.0. |
| Description | Sets the lower limit for the audio deviation measurement. The physical dimension of the number is stated as a Hz. |
| Query | There is query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:DEV:AUD:LIM:LOWer 0 :CALC:AMPS:RFTX:AUD:DEV:LIM:LOWer? Returns 0 in this example. |

:CALC:AMPS:RFTX:FREQ:ERR:RF:LIMit[:FAIL]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:RF:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:AMPS:ARR:RFTX:FREQ:ERR:RF measurement violated its limits. |
| Example | :MEAS:AMPS:ARR:RFTX:FREQ:ERR:RF 20 :CALC:AMPS:RFTX:FREQ:ERR:RF:LIM? Returns 0 if all 20 results are within limits, or 1 if any result is outside the limits. |

:CALC:AMPS:RFTX:FREQ:ERR:RF:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:RF:LIMit:STATe <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. The Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either ON or OFF. |
| Query | There is no query form of this command available. |
| Example | :CALC:AMPS:RFTX:FREQ:ERR:RF:LIM:STAT ON Switches the limit check for the frequency error measurement on. |

:CALC:AMPS:RFTX:FREQ:ERR:RF:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:RF:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is -5000, the maximum is 5000, the resolution is 1, and the default is value is 2600. |
| Description | Sets the upper limit for the frequency error measurement. The physical dimension of the number is stated as Hz. |
| Query | There is query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:FREQ:ERR:RF:LIM :UPP 2600 :CALC:AMPS:RFTX:FREQ:ERR:RF:LIM:UPP? Returns 2600 |

:CALC:AMPS:RFTX:FREQ:ERR:RF:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:RF:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is -5000, the maximum is 5000, the resolution is 1, the default is value is -2060. |
| Description | Sets the lower limit for the frequency error measurement. The physical dimension of the number is stated as Hz. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:FREQ:ERR:RF:LIM:LOWer 0 :CALC:AMPS:RFTX:FREQ:ERR:RF:LIM:LOWer? Returns 0 in this example. |

:CALC:AMPS:RFTX:FREQ:ERR:SAT:LIMit[:FAIL]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:SAT:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:AMPS:ARR:RFTX:FREQ:ERR:SAT measurement violated its limits. |

| | |
|----------------|---|
| Example | <pre>:MEAS:AMPS:ARR:RFTX:FREQ:ERR:SAT 20 :CALC:AMPS:RFTX:FREQ:ERR:SAT:LIM? Returns 0 if all 20 results are within limits, or 1 if any result is outside the limits.</pre> |
|----------------|---|

:CALC:AMPS:RFTX:FREQ:ERR:SAT:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:SAT:LIMit:STATe <PreDefExp> |
| Parameters | PreDefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either ON or OFF. |
| Query | There is no query form of this command available. |
| Example | <pre>:CALC:AMPS:RFTX:FREQ:ERR:SAT:LIM:STAT ON Switches the limit check for the SAT frequency error measurement on.</pre> |

:CALC:AMPS:RFTX:SAT:FREQ:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:SAT:FREQ:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is -5.0, the maximum is 5.0, the resolution is 1.0, and the default is value is 1.0. |
| Description | Sets the upper limit for the SAT frequency error measurement. The physical dimension of the number is stated as a Hz. |
| Query | There is a query form which returns the stored settings. |
| Example | <pre>:CALC:AMPS:RFTX:SAT:FREQ:LIM:UPP 200 :CALC:AMPS:RFTX:SAT:FREQ:LIM:UPP? Returns 200.</pre> |

:CALC:AMPS:RFTX:FREQ:ERR:SAT:LIMit:LOWer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:SAT:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is -5.0, the maximum is 5.0, the resolution is 1.0, the default is value is -1.0. |
| Description | Sets the lower limit for the SAT frequency error measurement. The physical dimension of the number is stated as a Hz. |
| Query | There is a query form which returns the stored settings. |
| Example | <pre>:CALC:AMPS:RFTX:FREQ:ERR:SAT:LIM:LOWer -1 :CALC:AMPS:RFTX:FREQ:ERR:SAT:LIM:LOWer? Returns -1 in this example.</pre> |

:CALC:AMPS:RFTX:FREQ:ERR:ST:LIMit[:FAIL]

| | |
|-------------------|---|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:ST:LIMit[:FAIL]? |
| Parameters | There are no parameters. |

| | |
|--------------------|--|
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:AMPS:ARR:RFTX:FREQ:ERR:ST measurement violated its limits. |
| Example | :MEAS:AMPS:ARR:RFTX:FREQ:ERR:ST 20 :CALC:AMPS:RFTX:FREQ:ERR:ST:LIM? Returns 0 if all 20 results are within limits, or 1 if any result is outside the limits. |

:CALC:AMPS:RFTX:FREQ:ERR:ST:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:ST:LIMit:STATe <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either ON or OFF. |
| Query | There is no query form of this command available. |
| Example | :CALC:AMPS:RFTX:FREQ:ERR:ST:LIM:STAT ON Switches the limit check for the ST frequency error measurement on. |

:CALC:AMPS:RFTX:FREQ:ERR:ST:LIMit:UPPer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:ST:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is -5.0, the maximum is 5.0, the resolution is 1.0, and the default is value is 1.0. |
| Description | Sets the upper limit for the ST frequency error measurement. The physical dimension of the number is stated as a Hz. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:AMPS:RFTX:FREQ:ERR:ST:LIM:UPP 1 :CALC:AMPS:RFTX:FREQ:ERR:ST:LIM:UPP? Returns 1. |

:CALC:AMPS:RFTX:FREQ:ERR:ST:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFTX:FREQ:ERR:ST:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. Its minimum value is -5.0, the maximum is 5.0, the resolution is 1.0, and the default is value is 1.0. |
| Description | Sets the lower limit for the ST frequency error measurement. The physical dimension of the number is stated as a Hz. |
| Query | :CALC:AMPS:RFTX:FREQ:ERR:ST:LIM:LOWer -1 :CALC:AMPS:RFTX:FREQ:ERR:ST:LIM:LOWer? Returns -1 in this example. |
| Example | :CALC:AMPS:RFTX:FREQ:ERR:ST:LIM:LOWer 0 :CALC:AMPS:RFTX:SAT:FREQ:LIM:LOWer? Returns 0 in this example. |

:CALC:AMPS:RFRX:SIN:LIMit[:FAIL]?

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFRX:SIN:LIMit[:FAIL]? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the :MEAS:AMPS:ARR:RFRX:SIN measurement violated its limits. |
| Example | :MEAS:AMPS:ARR:RFRX:SIN :CALC:AMPS:RFRX:SIN:LIM? Returns a 0 if any of the Sinad measurement results is out of limits. |

:CALC:AMPS:RFRX:SIN:LIMit:STATe

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFRX:SIN:LIMit:STATe <PredefExp> |
| Parameters | PredefExp may take on one of the following predefined expressions: ON OFF. The default is ON. |
| Description | This command switches the check of the measurement result values of the measurement against their user-definable limits either ON or OFF. |
| Query | There is no query form of this command available. |
| Example | :CALC:AMPS:RFRX:SIN:LIM:STAT ON Switches the limit check for the Sinad measurement on. |

:CALC:AMPS:RFRX:SIN:LIMit:UPPer:[DATA]

| | |
|--------------------|--|
| Syntax | :CALC:AMPS:RFRX:SIN:LIMit:UPPer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 0.0, the maximum is 100.0, the resolution is 0.1, the default is value is 100.0. |
| Description | Sets the upper limit for the Sinad measurement. The physical dimension of the number stated is dB. |
| Query | There is a query form which returns the stored settings. |
| Example | :CALC:AMPS:RFRX:SIN:LIM:UPP 20.00 :CALC:AMPS:RFRX:SIN:LIM:UPP? The query returns the previously set value 20.0. |

:CALC:AMPS:RFRX:SIN:LIMit:LOWer:[DATA]

| | |
|--------------------|---|
| Syntax | :CALC:AMPS:RFRX:SIN:LIMit:LOWer:[DATA] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for parameter is 0.0, the maximum is 100.0, the resolution is 0.1, the default is value is 12.0. |
| Description | Sets the lower limit for the Sinad measurement. The physical dimension of the number stated is dB. |
| Query | There is a query form which returns the stored settings. |

| | |
|----------------|---|
| Example | <pre>:CALC:AMPS:RFRX:SIN:LIM:LOW 10.0 :CALC:AMPS:RFRX:SIN:LIM:LOW? The query returns the previously set value 10.0.</pre> |
|----------------|---|

READ subsystem

The commands of this subsystem are used to read out the next measurement result values from the continuous measurements.

:READ:AMPS:RFTX:POWer

| | |
|--------------------|--|
| Syntax | <code>:READ:AMPS:RFTX:POWer?</code> |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the <code>:MEASure:AMPS:CONT ARRay:RFTX:POWer</code> measurement command. |
| Example | <code>:READ:AMPS:RFTX:POW?</code> |

:READ:CDMA:RFTX:POWer

| | |
|--------------------|--|
| Syntax | <code>:READ:CDMA:RFTX:POWer?</code> |
| Parameters | There are no parameters. |
| Description | Only the query form is supported. |
| Query | The command returns the results from the <code>:MEASure:CDMA:CONT ARRay:RFTX:POWer</code> measurement command. |
| Example | <code>:READ:CDMA:RFTX:POWer?</code> |

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 functionality 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 345](#).

- The RF generator and analyzer are enabled using the appropriate :CONFigure:CSYSTem command.

:RFGenerator:CDMA:STATE

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA: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 is in RF generator mode, there is no signaling active. This means that Wavetek 4400 may be used as RF signal generator for all kinds of RF and CDMA signals. While working as an RF generator, 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:CDMA:STATE ON :RFG:STAT? Value returned: "ON". |

:RFGenerator:CDMA:LEVel

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:LEVel <real1> |
| Parameters | real1 is a floating point real number. The minimum value for real1 is -120.0, the maximum value is 0.0. The minimum resolution for real1 is 0.1. The default value for real1 is -70.0. |
| Description | This command sets the RF output power level of 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:CDMA:LEVel -50.5 :RFG:CDMA:LEV? Value returned: "-50.5". |

:RFGenerator:CDMA:FREQuency

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:FREQuency <real1> |
| Parameters | real1 is a floating point real number. The minimum value for real1 is 430,000,000, the maximum value is 2,300,000,000. The minimum resolution for real1 is 10. The default value for real1 is 800,000,000. |
| Description | This command sets the RF generator's frequency. The value specified for real1 is the output frequency in Hz. |
| 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:CDMA:FREQuency 880200010 :RFG:CDMA:FREQ? Value returned: "880200010". |

:RFGenerator:CDMA:EXpected

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:EXpected <real1> |
| Parameters | real1 is a floating point real number. The minimum value for real1 is -70.0, the maximum value is 36.0. The minimum resolution for real1 is 0.1. The default value for real1 is 0.0. |
| Description | This command sets the expected power of the mobile. The value specified for real1 is the power 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:CDMA:EXpected -50.5 :RFG:CDMA:EXP? Value returned: "-50.5". |

:RFGenerator:CDMA:MODulation

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:MODulation <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: CDMA NONE. Default is CDMA. |
| 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 QPSK means that the output signal will be QPSK (Gaussian Minimum Shift Keying) modulated. Note: As soon as QPSK is selected, the Willtek 4400 will provide a CDMA-like RF output signal. |
| Query | The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above. |
| Example | :RFGenerator:CDMA:MODulation CDMA :RFG:CDMA:MOD? Value returned in this example "CDMA". |

:RFG:CDMA:BS:CCH:ARFCn

| | |
|--------------------|--|
| Syntax | :RFG:CDMA:BS:CCH:ARFCn <Int1>, [<PredefExp>] |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 2047. The default value is 29. PredefExp is an optional expression which can take on one of the following values: US800 USPCS TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. The default is US800. |
| Description | This command sets RF generator's frequency. Channel and band are used to calculate the frequency. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | RFG:CDMA:BS:CCH:ARFC 0,US800 |

:RFGenerator:CDMA:SSECTOR:[LEVEL]

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:SSECTOR:[LEVEL] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value is -12.0, the maximum value 3.0. The minimum resolution is 0.1. The default value is -3.0. |
| Description | This command sets the sector ratio. The physical dimension is dB. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFGenerator:CDMA:SSECTOR:LEV -7.0 :RFG:CDMA:SSECT? Value returned: -7.0. |

:RFGenerator:CDMA:SSECTOR:STATE

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:SSECTOR:STATE <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: OFF ON. Default is OFF. |
| Description | This command switches the sector 2 to OFF or ON. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFG:CDMA:SSEC:STATE ON :RFG:CDMA:SSEC:STAT? Value returned: ON. |

:RFGenerator:CDMA:SSECTOR:TCODE

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:SSECTOR:TCODE <Int1> |
| Parameters | Int1 is an integer number. The minimum value is 2, the maximum value 63. The default value is 8. |
| Description | This command sets the traffic code for sector 2. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFG:CDMA:SSECT:TCOD 8 :RFG:CDMA:SSECT:TCOD? Value returned: 8. |

:RFGenerator:CDMA:PILOT1:[LEVEL]

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:PILOT1:[LEVEL] <Reall> :RFGenerator:CDMA:PILOT2:[LEVEL] <Reall> |
| Parameters | Reall is a floating point real number. The minimum value is -32.0, the maximum value -5.0. The minimum resolution is 0.1. The default value is -7.0. |
| Description | This command sets the pilot channel power for sector 1 or sector 2. The physical dimension is dB. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |

| | |
|----------------|--|
| Example | <pre>:RFG:CDMA:PIL1:LEV -7.0 :RFG:CDMA:PIL1:LEV? Value returned for sector 1 pilot: -7.0. :RFG:CDMA:PIL2:LEV -16.0 :RFG:CDMA:PIL2:LEV? Value returned for sector 2 pilot: -16.0.</pre> |
|----------------|--|

:RFGenerator:CDMA:PILOt1:ECIO

| | |
|--------------------|---|
| Syntax | <pre>:RFGenerator:CDMA:PILOt1:ECIO? :RFGenerator:CDA:PILOt2:ECIO?</pre> |
| Parameters | There are no parameters. |
| Description | Pilot Channel Power to Pilot Channel Power plus Noise Ratio expressed in dB. There is solely a query form of this command available. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one real number. |
| Example | <pre>:RFG:CDMA:PIL1:ECIO? Value returned for sector 1: -7.0. :RFG:CDMA:PIL2:ECIO? Value returned for sector 2: -7.0.</pre> |

:RFGenerator:CDMA:SYNC:[LEVEL]

| | |
|--------------------|---|
| Syntax | <pre>:RFGenerator:CDMA:SYNC:[LEVEL] <Reall></pre> |
| Parameters | Reall is a floating point real number. The minimum value is -32.0, the maximum value -5.0. The minimum resolution is 0.1. The default value is -16.0. |
| Description | This command sets the sync channel power. The physical dimension is dB. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | <pre>:RFG:CDMA:SYNC:LEV -7.0 :RFG:CDMA:SYNC:LEV? Value returned for: -7.0.</pre> |

:RFGenerator:CDMA:SYNC:ECIO

| | |
|--------------------|---|
| Syntax | <pre>:RFGenerator:CDMA:SYNC:ECIO?</pre> |
| Parameters | There are no parameters. |
| Description | Sync. Channel Power to Sync Channel Power plus Noise Ratio expressed in dB. There is solely a query form of this command available. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one real number. |
| Example | <pre>:RFG:CDMA:SYNC:ECIO? Value returned for sector 1: -7.0.</pre> |

:RFGenerator:CDMA:SSECTOR:[LEVEL]

| | |
|-------------------|---|
| Syntax | <pre>:RFGenerator:CDMA:SSECTOR:[LEVEL] <Int1></pre> |
| Parameters | Int1 is an integer number. The minimum value is -5, the maximum value 2. The default value is -5. |

| | |
|--------------------|---|
| Description | This command sets the quick paging channel power. The physical dimension is dB. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFGenerator:CDMA:QPCHannel:LEV -3 :RFG:CDMA:QPCHannel:LEV? Value returned: -3. |

:RFGenerator:CDMA:QPCH:STATE

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:QPCH:STATE <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: OFF ON. Default is OFF. |
| Description | This command switches the generation of the quick paging channel to OFF or ON. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFG:CDMA:QPCH:STAT ON :RFG:CDMA:QPCH:STAT? Value return: ON. |

:RFGenerator:CDMA:SSECTOR:[LEVEL]

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:SSECTOR:[LEVEL] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is -32.0, the maximum value -5.0. The minimum resolution is 0.1. The default value is -12.0. |
| Description | This command sets the sector ratio. The physical dimension is dB. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFGenerator:CDMA:SSECTOR:LEV -7.0 :RFG:CDMA:SSECT? Value returned: -7.0. |

:RFGenerator:CDMA:PAG:ECIO

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:PAG:ECIO? |
| Parameters | There are no parameters. |
| Description | Paging channel power to paging channel power plus noise ratio expressed in dB. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFG:CDMA:PAGing:ECIO? Value returned for sector 1: -7.0. |

:RFGenerator:CDMA:FCH1:[LEVEL]

| | |
|-------------------|--|
| Syntax | :RFGenerator:CDMA:FCH1:[LEVEL] <Real1> :RFGenerator:CDMA:FCH2:[LEVEL] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is -32.0, the maximum value -5.0. The minimum resolution is 0.1. The default value is -7.4. |

| | |
|--------------------|--|
| Description | This command sets the fundamental channel power for sector 1 or sector 2. The physical dimension is dB. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFG:CDMA:FCH1:LEV -7.5 :RFG:CDMA:FCH1:LEV? Value returned for sector 1 Fundamental power: -7.5. :RFG:CDMA:FCH2:LEV -6.2 :RFG:CDMA:FCH2:LEV? Value returned for sector 2 Fundamental power: -6.2. |

:RFGenerator:CDMA:FCH1:ECIO

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:FCH1:ECIO? :RFGenerator:CDMA:FCH2:ECIO? |
| Parameters | There are no parameters. |
| Description | Fundamental channel power to fundamental channel power plus noise ratio expressed in dB. There is solely a query form of this command available. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one real number. |
| Example | :RFG:CDMA:FCH1:ECIO? Value returned for sector 1 for example: -7.4. :RFG:CDMA:FCH2:ECIO? Value returned for sector 2 for example: -7.4. |

:RFGenerator:CDMA:FCH1:EBNT

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:FCH1:EBNT? :RFGenerator:CDMA:FCH2:EBNT? |
| Parameters | There are no parameters. |
| Description | Ratio in dB of the fundamental channel's combined received energy per bit to the effective noise power spectral density. There is solely a query form of this command available. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFG:CDMA:FCH1:EBNT? Value returned for sector 1 for example: -7.4. :RFG:CDMA:FCH2:EBNT? Value returned for sector 2 for example: -7.4. |

:RFGenerator:CDMA:FCH1:RATE

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:FCH1:RATE? :RFGenerator:CDMA:FCH2:RATE? |
| Parameters | There are no parameters. |
| Description | Only the query form is available. |
| Query | This query returns the data rate of fundamental channels in sector 1 or sector 2. |
| Example | :RFG:CDMA:FCH1:RATE? Value returned for sector 1: FULL. :RFG:CDMA:FCH2:RATE? Value returned for sector 2: FULL. |

:RFGenerator:CDMA:SCH:[LEVEL]

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:SCH:[LEVEL] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is -32.0, the maximum value -5.0. The minimum resolution is 0.1. The default value is -7.4. |
| Description | This command sets the sector ratio. The physical dimension is dB. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFGenerator:CDMA:SSECTOR:LEV -7.4 :RFG:CDMA:SSECT? Value returned: -7.4. |

:RFGenerator:CDMA:SCH:ECIO

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:SCH:ECIO? |
| Parameters | There are no parameters. |
| Description | Supplemental channel power to supplemental channel power plus noise ratio expressed in dB. There is solely a query form of this command available. |
| Query | The query form of this command will return the current setting of the Willtek 4400. The string delivered back will contain one real number. |
| Example | :RFG:CDMA:SCH:ECIO? Value returned for sector 1: -7.0. |

:RFGenerator:CDMA:SCH:EBNT?

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:SCH:EBNT? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | Returns the ratio in dB of the supplemental channel's combined received energy per bit to the effective noise power spectral density. |
| Example | :RFG:CDMA:SCH:EBNT? Value returned for example: -7.4. |

:RFGenerator:CDMA:SCH:RATE

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:SCH:RATE? |
| Parameters | There are no parameters. |
| Description | The command form is not available. |
| Query | This query returns the data rate of supplemental channel. |
| Example | :RFG:CDMA:SCH:RATE? Value returned for example: FULL. |

:RFGenerator:CDMA:OCNS1:[LEVEL]

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:OCNS1:[LEVEL]? :RFGenerator:CDMA:OCNS2:[LEVEL]? |
| Parameters | There are no parameters. |
| Description | The command form is not available. |
| Query | This query returns the OCNS channel power for sector 1 or sector 2. The physical dimension is dB. |
| Example | :RFG:CDMA:OCNS1:LEV? Value returned for sector 1 OCNS, for example -4.5. :RFG:CDMA:OCNS2:LEV? Value returned for sector 2 OCNS, for example -2.1. |

:RFGenerator:CDMA:FCH1:ECIO

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:FCH1:ECIO? :RFGenerator:CDMA:FCH2:ECIO? |
| 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 ratio of OCNS power to OCNS power plus noise, expressed in dB. |
| Example | :RFG:CDMA:OCNS1:ECIO? Value returned for sector 1, for example: -6.2. :RFG:CDMA:OCNS2:ECIO? Value returned for sector 2, for example: -6.8. |

:RFGenerator:CDMA:AWGN:[LEVEL]

| | |
|--------------------|--|
| Syntax | :RFGenerator:CDMA:AWGN:[LEVEL] <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value is -10.0, the maximum value 5.0. The minimum resolution is 0.1. The default value is 0.0. |
| Description | This command sets the AWGN level. The physical dimension is dB. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFGenerator:CDMA:AWGN:LEV -1.0 :RFG:CDMA:AWGN:LEV? Value returned: -1.0. |

:RFGenerator:CDMA:AWGN:STATE

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:AWGN:STATE <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: OFF ON. Default is OFF. |
| Description | This command switches the generation of AWGN to OFF or ON. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFG:CDMA:AWGN:STAT ON :RFG:CDMA:AWGN:STAT? Value returned: ON. |

:RFGenerator:CDMA:TXCorrection[:IMMEDIATE]

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:TXCorrection[:IMMEDIATE] |
| Parameters | There are no parameters. |
| Description | This command starts the internal TX correction procedure. This procedure should be carried out when a temperature change has occurred since the last TX level correction, before performing a quality measurement. The need for such a procedure can be tested with the :RFG:CDMA:TXCO:CALibrate query. |
| Query | There is no query form of this command. |
| Example | :RFG:CDMA:TXCO |

:RFGenerator:CDMA:TXCorrection:CALibrate

| | |
|--------------------|---|
| Syntax | :RFGenerator:CDMA:TXCorrection:CALibrate? |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is OFF. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers a boolean number in a string which indicates if the calibration command should be issued. |
| Example | :RFG:CDMA:TXCO:CALi? The query returns 1 if the RFG:CDMA:TXCO command needs to be issued, or 0 if ok. |

:RFGenerator:AMPS:STATE

| | |
|--------------------|--|
| Syntax | :RFGenerator:AMPS:STATE <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is OFF. |
| Description | Switches the RF generator ON or OFF. While the Willtek 4400 is in RF generator mode, there is no signalling active. This means that the Willtek 4400 may be used as RF signal generator for all kinds of RF and CDMA signals. While working as a RF generator, the 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:AMPS:STATE ON :RFG:AMPS:STAT? Value returned in this example "ON". |

:RFGenerator:AMPS:LEVEL

| | |
|---------------|---------------------------------|
| Syntax | :RFGenerator:AMPS:LEVEL <real1> |
|---------------|---------------------------------|

| | |
|--------------------|---|
| Parameters | <code>real1</code> is a floating point real number. The minimum value for <code>real1</code> is <code>-120.0</code> , the maximum value is <code>0.0</code> . The minimum resolution for <code>real1</code> is <code>0.1</code> . The default value for <code>real1</code> is <code>-70.0</code> . |
| Description | This command sets the RF output power level of Willtek 4400. The value specified for <code>real1</code> 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 | <code>:RFGenerator:AMPS:LEVel -50.5</code> <code>:RFG:AMPS:LEV?</code> Value returned in this example: <code>"-50.5"</code> . |

:RFGenerator:AMPS:FREQuency

| | |
|--------------------|---|
| Syntax | <code>:RFGenerator:AMPS:FREQuency <real1></code> |
| Parameters | <code>real1</code> is a floating point real number. The minimum value for <code>real1</code> is <code>430,000,000</code> , the maximum value is <code>2,300,000,000</code> . The minimum resolution for <code>real1</code> is <code>10</code> . The default value for <code>real1</code> is <code>800,000,000</code> . |
| Description | This command sets the RF generator's frequency. The value specified for <code>real1</code> is the output frequency in Hz. Note: For more details on the CDMA frequency ranges, please refer to section Basic Specifications of CDMA 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 | <code>:RFGenerator:AMPS:FREQuency 880200010</code> <code>:RFG:CDMA:FREQ?</code> Value returned in this example: <code>"880200010"</code> . |

:RFG:AMPS:BS:VCH:ARFCn

| | |
|--------------------|--|
| Syntax | <code>:RFG:AMPS:BS:VCH:ARFCn <Int1>, [<PredefExp>]</code> |
| Parameters | <code>Int1</code> is an integer number. The minimum value is <code>0</code> , the maximum value <code>2047</code> . The default value is <code>29</code> . <code>PredefExp</code> is an optional expression which can take on one of the following values: <code>US800</code> <code>USPCs</code> <code>TACS</code> <code>JTACS</code> <code>KPCS</code> <code>NMT450</code> <code>IMT2000</code> <code>MHZ900</code> <code>MHZ1800</code> . The default is <code>US800</code> . |
| Description | This command sets RF generator's frequency. The channel and band are used to calculate the frequency. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | <code>:RFG:AMPS:BS:VCH:ARFC 0,AMPS</code> |

:RFGenerator:AMPS:MODulation

| | |
|---------------|---|
| Syntax | <code>:RFGenerator:AMPS:MODulation <PredefExp></code> |
|---------------|---|

| | |
|--------------------|--|
| Parameters | PredefExp is one of the following predefined expressions: FM NONE. Default is FM. |
| Description | This command selects the type of modulation to be used on the RF output signal. NONE means that the output signal will not be modulated, while FM means that the output signal will be frequency modulated. |
| 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:AMPS:MODulation FM :RFG:AMPS:MOD? Value returned in this example "FM". |

:RFGenerator:AMPS:MOD:FM:SAT

| | |
|--------------------|--|
| Syntax | :RFGenerator:AMPS:MOD:FM:SAT <int1> |
| Parameters | int1 is one of the following predefined expressions: 0,1,2. Default is 1. |
| Description | This command defines which FM modulation rate will be utilized to modulate the output signal. 0 = 5970 Hz, 1 - 6000 Hz, 2 - 6030Hz. |
| Query | The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions |
| Example | :RFGenerator:AMPS:MOD:FM:SAT 0 :RFG:AMPS:MOD:FM:SAT? Value returned in this example "0". |

:RFGenerator:AMPS:MOD:FM:KHZ

| | |
|--------------------|---|
| Syntax | :RFGenerator:AMPS:MOD:FM:KHZ <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: ON OFF. Default is ON. |
| Description | This command enables or disables the 1kHz tone. |
| 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:AMPS:MOD:FM:KHZ ON :RFG:AMPS:MOD:FM:KHZ? Value returned in this example "0". |

:RFGenerator:AMPS:TXCorrection [:IMMEDIATE]

| | |
|--------------------|---|
| Syntax | :RFGenerator:AMPS:MOD:FM:KHZ [:IMMEDIATE] |
| Parameters | There are no parameters. |
| Description | This command sends a message to perform TX Correction procedure |
| Query | There is no query form of this command. |
| Example | :RFG:AMPS:TXC. |

:RFG:AMPS:TXC:CAL

| | |
|--------------------|---|
| Syntax | :RFG:AMPS:TXC:CAL? |
| Parameters | There are no parameters. |
| Description | There is solely a query form of this command available. |
| Query | This command delivers boolean number in a string which indicates if Calibration Command should be issued. |
| Example | :RFG:AMPS:TXC:CAL? The query returns 1, if the RFG:CDMA:TXC command needs to be issued, or 0 if ok. |

RFANalyser subsystem

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

:RFAN:CDMA:BS:CCH:ARFCn

| | |
|--------------------|--|
| Syntax | :RFAN:CDMA:BS:CCH:ARFCn <Int1>, [<PredefExp>] |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 2047. The default value is 29. PredefExp is an optional expression which can take on one of the following values: US800 USPCS TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. The default is US800. |
| Description | This command sets RF generator's frequency. The channel and Band are used to calculate the frequency. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFAN:CDMA:BS:CCH:ARFC 0,US800 |

:RFANalyser:CDMA:FREQuency

| | |
|--------------------|--|
| Syntax | :RFANalyser:CDMA:FREQuency <Reall> |
| Parameters | Reall is a floating point real number. The minimum value for Reall is 430,000,000, the maximum value is 2,300,000,000. The minimum resolution for Reall is 10. The default value is 800,000,000. |
| Description | This command sets the RF analyzer's center frequency. The value specified is the center frequency in Hz. The frequency of the internal synthesizer can be changed in steps of 10 Hz. |
| Query | The query form of this command will return the current setting. The string delivered back will contain one floating point real number. |
| Example | :RFANalyser:CDMA:FREQuency 880200000 :RFAN:CDMA:FREQ? Value returned: 880200000. |

:RFANalyser:CDMA:ESN

| | |
|--------------------|--|
| Syntax | :RFANalyser:CDMA:ESN <string1> |
| Parameters | string1 is a string of up to 15 characters. |
| Description | This command sets the Electronic Serial Number (ESN) used by the RF analyzer for CDMA2000 waveform quality measurements. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFAN:CDMA:ESN AAAABBCD :RFAN:CDMA:ESN? Value returned: AAAABBCD. |

:RFANalyser:CDMA:RCONfig

| | |
|--------------------|---|
| Syntax | :RFANalyser:CDMA:RCONfig[:REVerse] <int1> |
| Parameters | int1 is an integer number. The minimum value is 1, the maximum value 4. The default value is 1. |
| Description | This command sets the reverse radio configuration used by the RF analyzer for CDMA2000 waveform quality measurements. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFAN:CDMA:RCON:REV 2 :RFAN:CDMA:RCON:REV? Value returned: 2. |

:RFANalyser:AMPS:FREQuency

| | |
|--------------------|--|
| Syntax | :RFANalyser:AMPS:FREQuency <Real1> |
| Parameters | Real1 is a floating point real number. The minimum value for Real1 is 430,000,000, the maximum value is 2,300,000,000. The minimum resolution for real1 is 10. The default value is 800,000,000. |
| Description | This command sets the RF analyzer's center frequency. The value specified is the center frequency in Hz. The frequency of the internal synthesizer can be changed in steps of 10 Hz. |
| Query | The query form of this command will return the current setting. The string delivered back will contain one floating point real number. |
| Example | :RFANalyser:AMPS:FREQuency 880200000 :RFAN:AMPS:FREQ? Value returned in this example: 880200000. |

:RFAN:AMPS:BS:VCH:ARFCn

| | |
|--------------------|--|
| Syntax | :RFAN:AMPS:BS:VCH:ARFCn <Int1>, [<PredefExp>] |
| Parameters | Int1 is an integer number. The minimum value is 0, the maximum value 2047. The default value is 29. PredefExp is an optional expression which can take on one of the following values: US800 USPCS TACS JTACS KPCS NMT450 IMT2000 MHZ900 MHZ1800. The default is US800. |
| Description | This command sets RF Analyzer's frequency. The channel and band are used to calculate the frequency. |
| Query | The query form of this command will return the current setting of the Willtek 4400. |
| Example | :RFAN:AMPS:BS:VCH:ARFC 0,AMPS |

RFSPepectrum subsystem

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

:RFSpectrum:CMDA:MSpectrum:SPAN

| | |
|--------------------|--|
| Syntax | :RFSpectrum:CMDA:MSpectrum:SPAN <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: SP500 SP1250 SP2500. Default is SP500. |
| Description | Sets the span (i.e. the spectral bandwidth) of the modulation spectrum analyzer. SP500 will set a span of 500 kHz, SP1250 will set a span of 1.25 MHz. SP2500 will set a span of 2.5 MHz. |
| Query | The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above. |
| Example | :RFSpectrum:CDMA:MSpectrum:SPAN SP1250 :RFSpectrum:CDMA:MSpectrum:SPAN? Value returned: "SP1250". |

:RFSpectrum:CDMA:MSpectrum:RESolution

| | |
|--------------------|---|
| Syntax | :RFSpectrum:CDMA:MSpectrum:RESolution <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: RES2 RES5 RES10 RES30. Default is RES2. |
| Description | Sets the spectral resolution of the modulation spectrum analyzer. RES2 stands for a resolution of 2 kHz. This resolution may only be chosen in conjunction with the 500 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 500 kHz and 1.25 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 2.5 MHz spectral bandwidth (span) as described above. Notes: <ul style="list-style-type: none"> - With the span set to 500 kHz, only the following spectral resolutions are available: 2 kHz (RES2), 5 kHz (RES5) and 10 kHz (RES10). - With the span set to 1.25 MHz, only the following spectral resolutions are available: 5 kHz (RES5) and 10 kHz (RES10). - With the span set to 2.5 MHz, only the following spectral resolutions are available: 10 kHz (RES10) and 30 kHz (RES30). - 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 | :RFSpectrum:CDMA:MSpectrum:RESolution RES10 :RFSpectrum:CDMA:MSpectrum:RESolution? Value returned: "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 | <p>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).</p> <p>Notes</p> <ul style="list-style-type: none"> – 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 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 | <pre>:AFGenerator:OUTPut CODEc :AFG:OUTP? Value returned: "COD".</pre> |

: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 | <p>This command sets the output voltage of the signals, available at the "AF out" connector, located on the front panel of the Willtek 4400.</p> <p>The value specified for real1 is the peak voltage of the AF signal.</p> <p>Notes</p> <ul style="list-style-type: none"> – 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 | <pre>:AFGenerator:LEVel[:RELative]:AF 2.5 :AFG:LEV:AF? Value returned in this example: "2.5".</pre> |

: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 | <p>This command sets the frequency of the AF generator as long as it is run in single-tone generation mode.</p> <p>The physical dimension of real1 is Hertz.</p> <p>Note: Please refer to the description of the :AFG:MODE command for further details regarding the single-tone generation mode.</p> |

| | |
|----------------|--|
| 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 efr1.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 <ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> - 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:BALEnced

| | |
|--------------------|---|
| Syntax | :AFANalyser:BALEnced <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:BALEnced OFF :AFAN:BALE? 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 | <p>This command sets the range of the expected input voltage on the Willtek 4400 "AF in" connector.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>MV300 expects a maximum input voltage of 300 mV (rms) and delivers a resolution of 1 mV, while</p> <p>MV30 means that the maximum input voltage will be 30 mV (rms) and the resolution will be 100 µV.</p> <p>Notes</p> <ul style="list-style-type: none"> – 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 | <pre>:AFANalyser:VRANge V30 :AFAN:VRAN? Value returned: "V30".</pre> |

:AFANalyser:MRANge

| | |
|--------------------|---|
| Syntax | :AFANalyser:MRANge <PredefExp> |
| Parameters | PredefExp is one of the following predefined expressions: AUTO V1 MV100 . Default is AUTO . |
| Description | <p>This command sets the range of the expected input voltage on the Willtek 4400 "AUX 4 (AUX in)" connector.</p> <p>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.</p> <p>V1 means that an input voltage of up to 1 Volt (rms) is expected and that the resolution will be 1 mV.</p> <p>MV100 expects a maximum input voltage of 100 mV (rms) and delivers a resolution of 100 µV.</p> <p>Notes</p> <ul style="list-style-type: none"> – 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 | <pre>:AFANalyser:MRANge MV100 :AFAN:MRAN? Value returned: "MV100".</pre> |

: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 <ul style="list-style-type: none"> - 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 | <p>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.</p> <p>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.</p> |
| Query | <p>The query form of this command returns the current setting. A query will return a string, containing one of the predefined expressions explained above.</p> |
| Example | <pre>:PSUPply:MEASure:CMAX MA400 :PSUP:MEAS:CMAX? Value returned: "MA400".</pre> |

Warranty and Repair



B

This chapter describes the customer services available through Willtek. Topics discussed in this chapter include the following:

- ["Warranty information" on page 530](#)
- ["Equipment return instructions" on page 531](#)

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 |
|------------|--|
| 0310-400-A | First 4400 software revision including CDMA2000. |
| 0406-410-A | AMPS included in CDMA2000 System Option. DHCP operation supported by TCP/IP Option. |
| 0411-500-A | New software version 5.00 that includes all different system options. No functional changes. |
| 0506-510-A | New software version 5.10. AMPS SCPI commands added. Updated test status messages for Call Mode. Performed some corrections in the RAPID chapter. |
| 0508-610-A | New software version 6.10. Screen shots updated. |
| 0602-620-A | New software version 6.20. |

Willtek and its logo are trademarks of Willtek Communications GmbH. All other trademarks and registered trademarks are the property of their respective owners.

Specifications, terms and conditions are subject to change without notice.

© Copyright 2006 Willtek Communications GmbH. All rights reserved.

No part of this manual may be reproduced or transmitted in any form or by any means (printing, photocopying or any other method) without the express written permission of Willtek Communications GmbH.

Manual ident no.
M 292 010
Manual version
0602-620-A
English

Willtek Communications GmbH
85737 Ismaning
Germany
Tel: +49 (0) 89 996 41-0
Fax: +49 (0) 89 996 41-440
info@willtek.com

Willtek Communications UK
Cheadle Hulme
United Kingdom
Tel: +44 (0) 161 486 3353
Fax: +44 (0) 161 486 3354
willtek.uk@willtek.com

Willtek Communications SARL
Roissy
France
Tel: +33 (0) 1 72 02 30 30
Fax: +33 (0) 1 49 38 01 06
willtek.fr@willtek.com

Willtek Communications Inc.
Parsippany
USA
Tel: +1 973 386 9696
Fax: +1 973 386 9191
willtek.cala@willtek.com
sales.us@willtek.com

Willtek Communications
Singapore
Asia Pacific
Tel: +65 943 63 766
willtek.ap@willtek.com

Willtek Communications Ltd.
Shanghai
China
Tel: +86 21 5835 8039
Fax: +86 21 5835 5238
willtek.cn@willtek.com

© Copyright 2006
Willtek Communications GmbH. All
rights reserved.
Willtek Communications, Willtek
and its logo are trademarks of
Willtek Communications GmbH. All
other trademarks and registered
trademarks are the property of their
respective owners.

Note: Specifications, terms and con-
ditions are subject to change with-
out prior notice.

will'tek